FIG.1
PRIOR ART

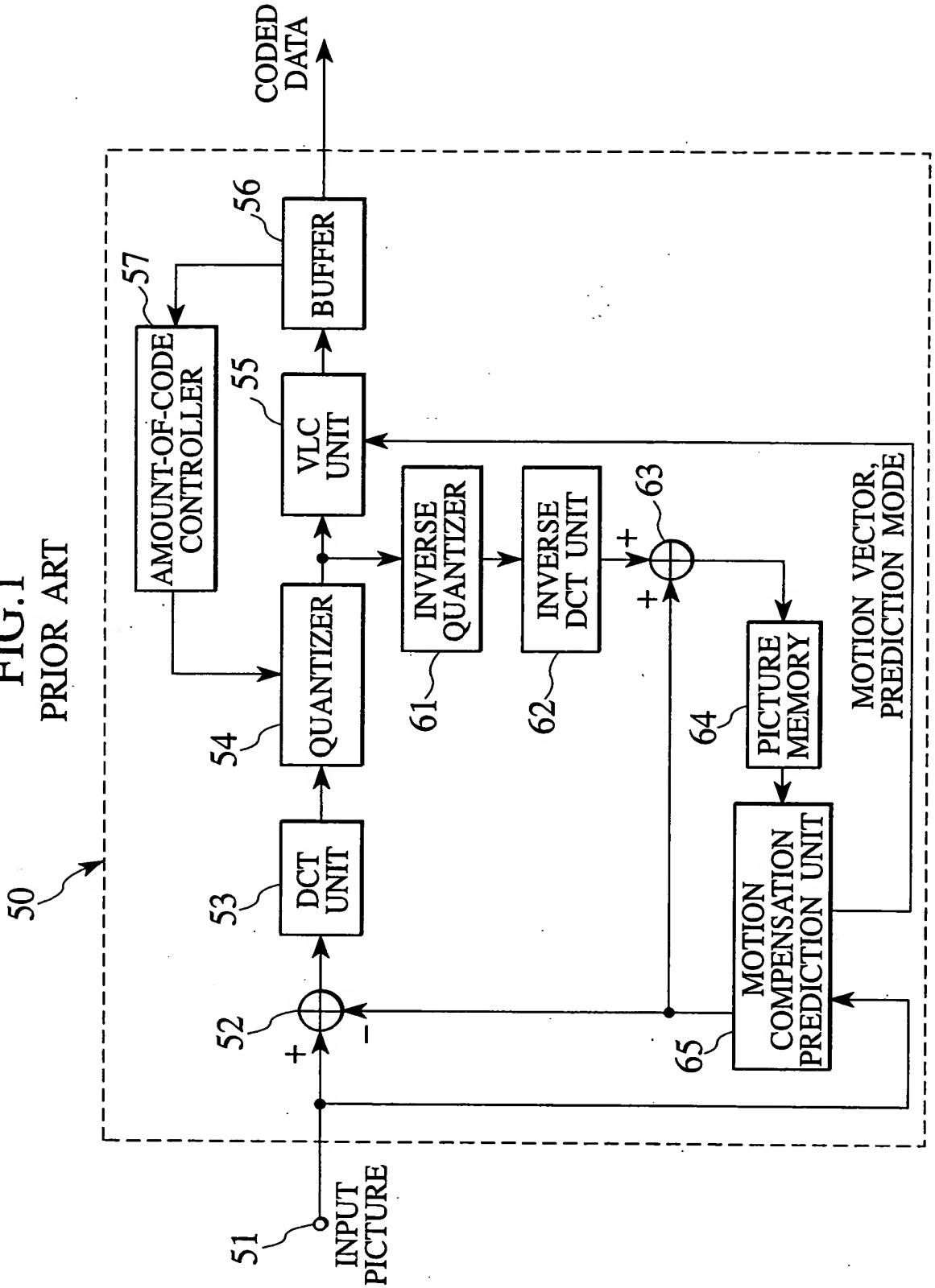


FIG.2
PRIOR ART

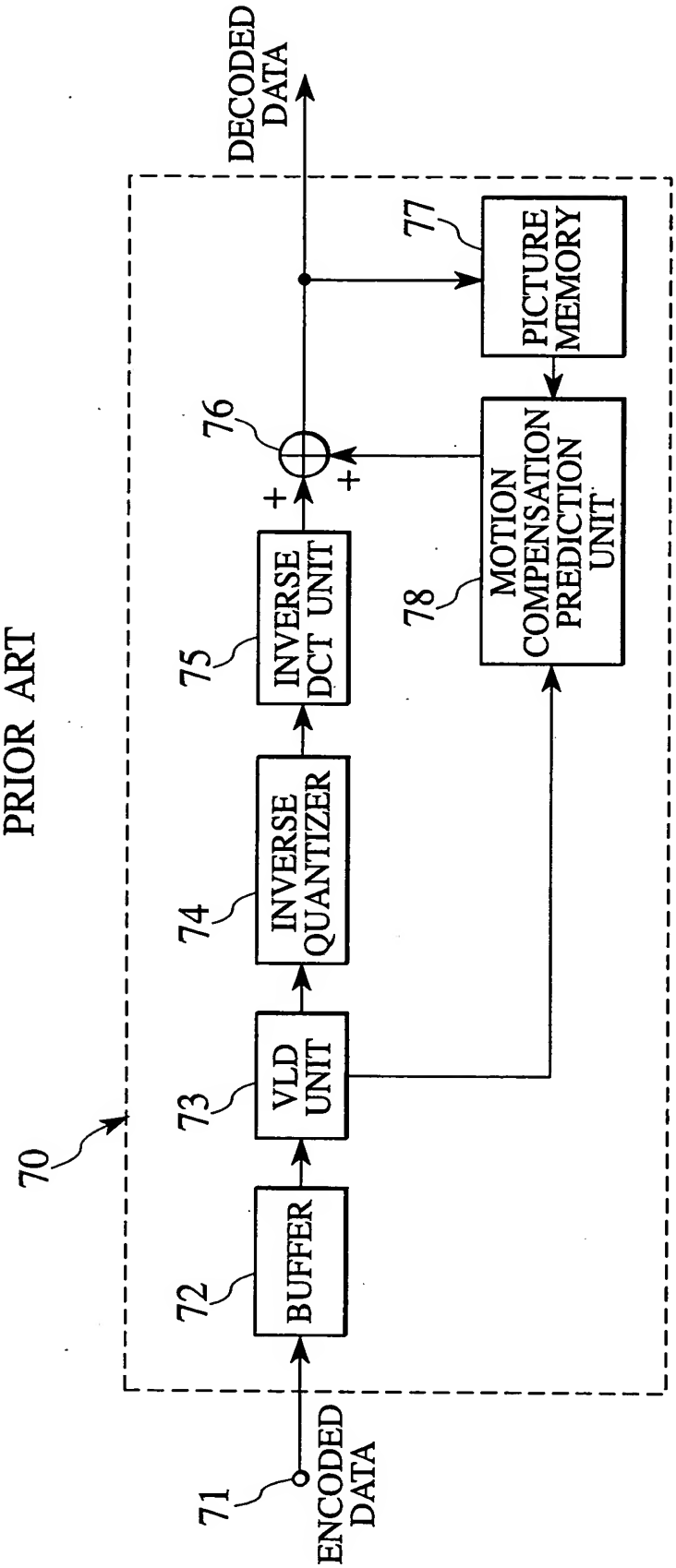


FIG.3
PRIOR ART

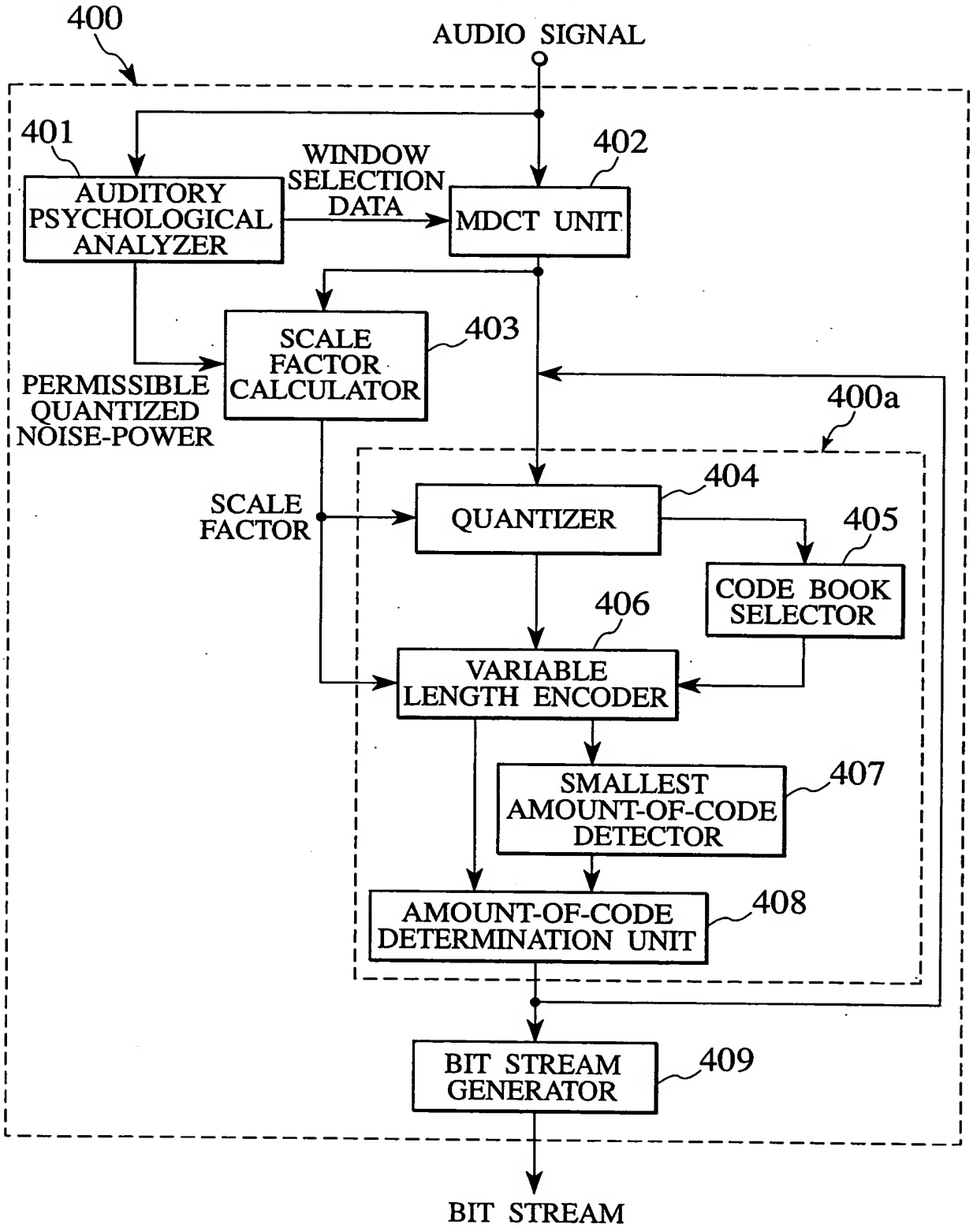


FIG.4
PRIOR ART

MAXIMUM QUANTIZATION VALUE	HUFFMAN CODE BOOK
$Q = 0$	0
$Q \leqq 1$	1 or 2
$Q \leqq 2$	3 or 4
$Q \leqq 4$	5 or 6
$Q \leqq 7$	7 or 8
$Q \leqq 12$	9 or 10
$Q > 12$	11 (ESC)

FIG.5
PRIOR ART

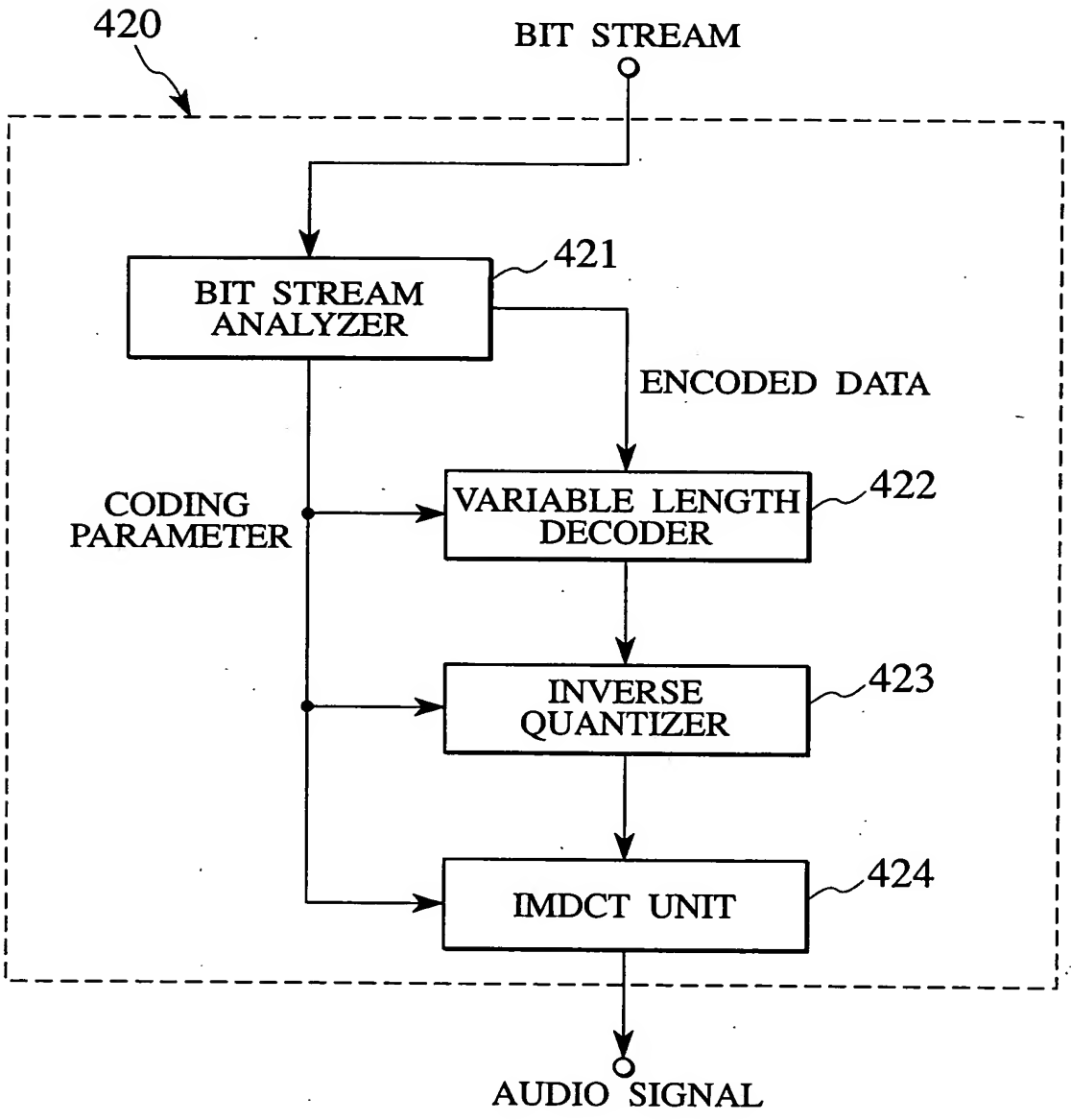


FIG.6

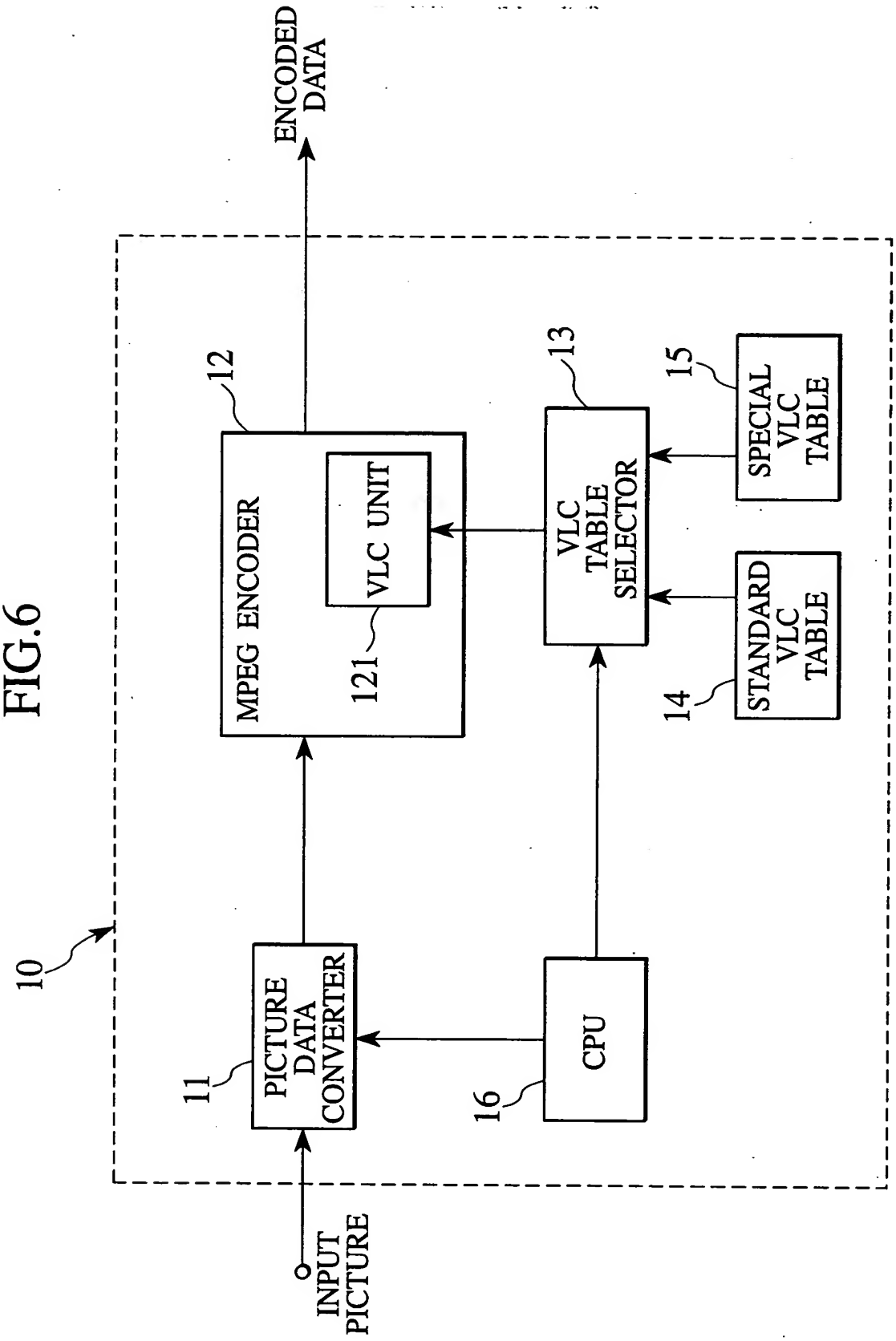

ENCODED DATA
12
MPEG ENCODER
VLC UNIT
121
13
VLC TABLE SELECTOR
15
SPECIAL VLC TABLE
14
STANDARD VLC TABLE
10
11
PICTURE DATA CONVERTER
16
CPU
INPUT PICTURE

FIG.7

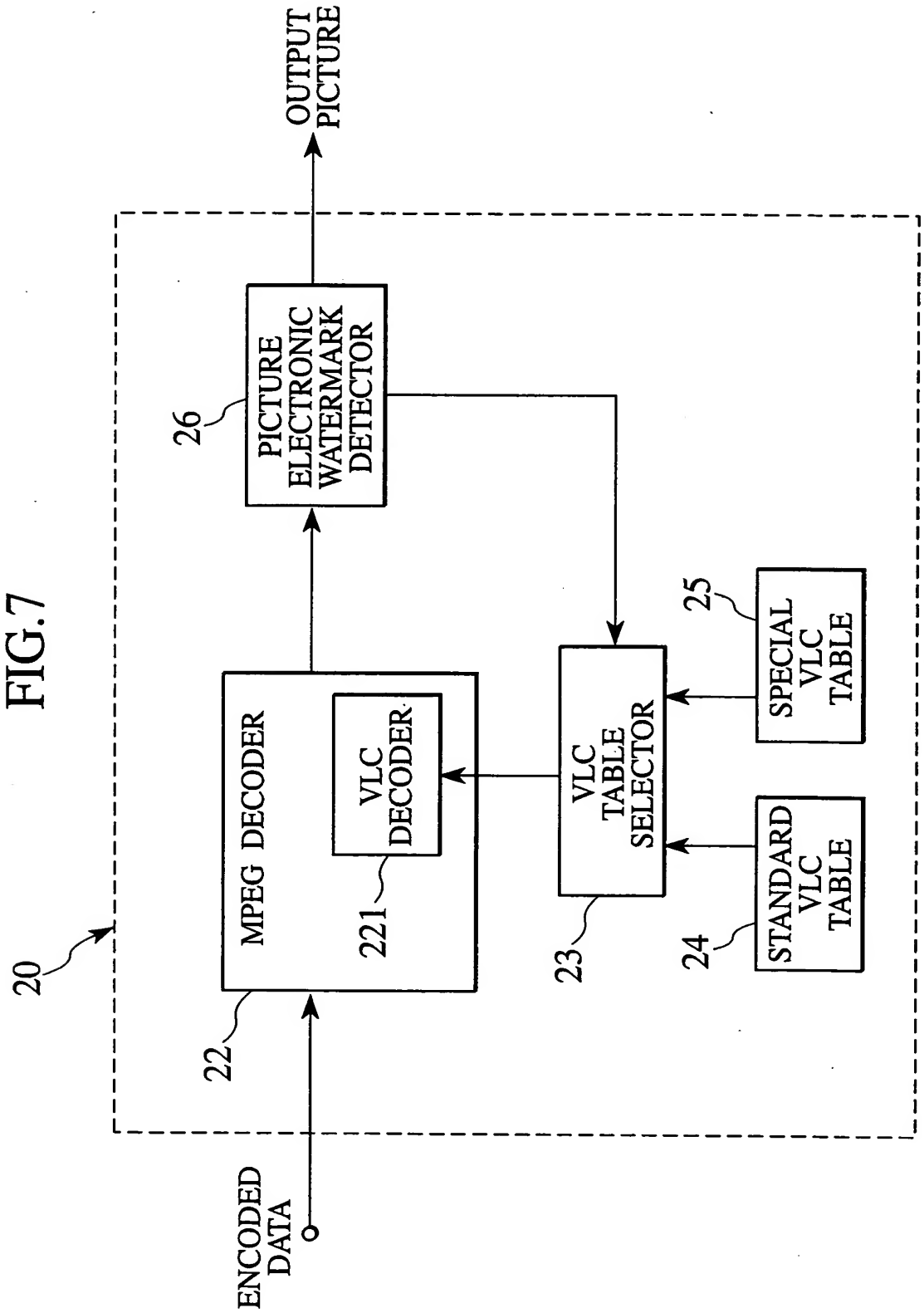

OUTPUT PICTURE
26
PICTURE ELECTRONIC WATERMARK DETECTOR
MPEG DECODER
VLC DECODER
221
22
20
ENCODED DATA
VLC TABLE SELECTOR
23
25
SPECIAL VLC TABLE
24
STANDARD VLC TABLE

FIG.8

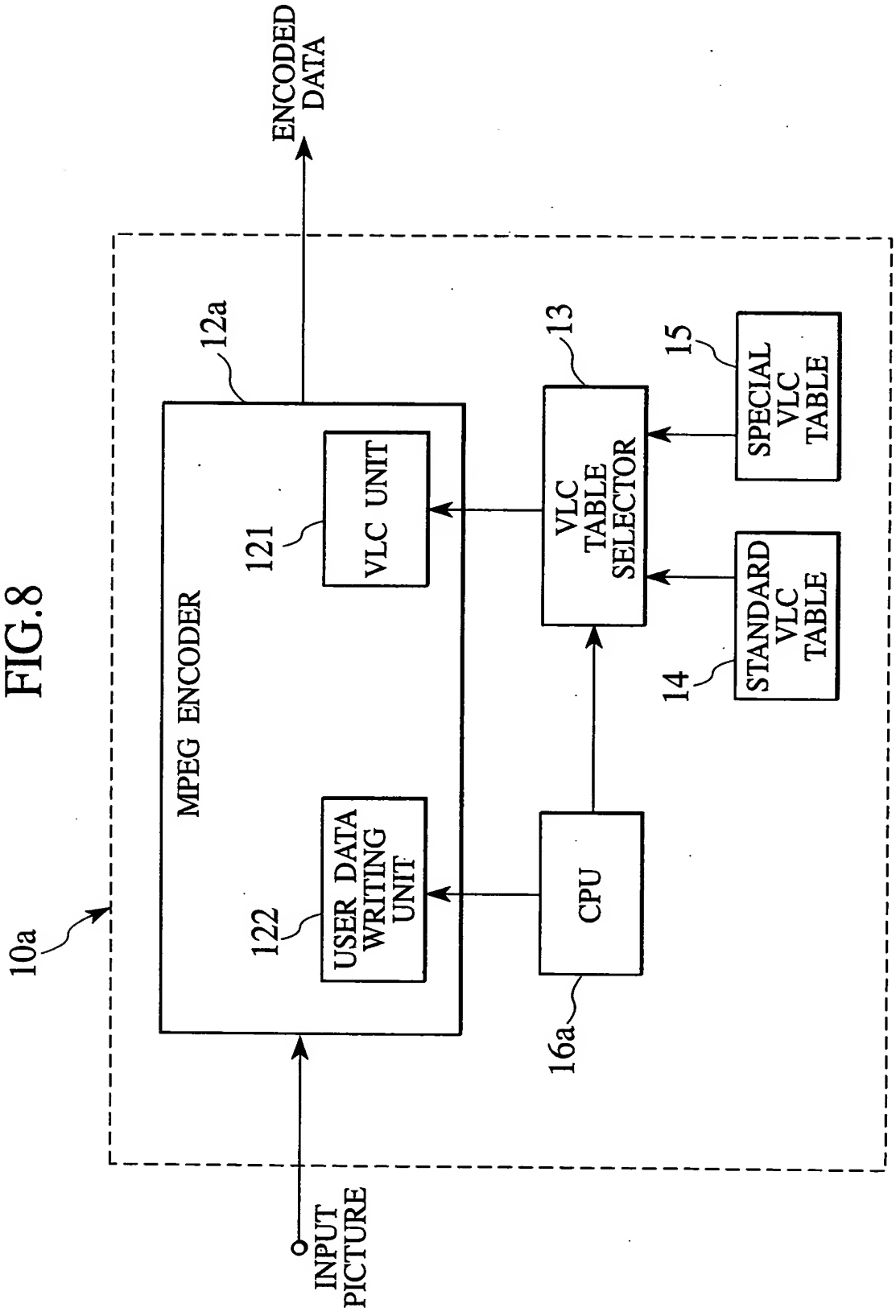

ENCODED DATA
12a
MPEG ENCODER
121
VLC UNIT
122
USER DATA WRITING UNIT
13
VLC TABLE SELECTOR
15
SPECIAL VLC TABLE
14
STANDARD VLC TABLE
16a
CPU
10a
INPUT PICTURE

FIG.9

Sequence header

SYNTAX	BIT NUMBER
Sequence_header( ) {	
Sequence_header_code	32
Horizontal_size	12
Vertical_size	12
Pel_aspect_ratio	4
Picture_rate	4
Bit_rate	18
Marker_bit	1
Vbv_buffer_size	10
Constrained_parameters_flag	1
Load_intra_quantizer_matrix	1
if (load_intra_quantizer_matrix)	
Intra_quantizer_matrix [ ]	8 * 64
Load_non_intra_quantizer_matrix	1
if (load_non_intra_quantizer_matrix)	
Non_intra_quantizer_matrix [ ]	8 * 64
Next_start_code( )	
if (nextbits( ) == extension_start_code) {	
Extension_start_code	32
While (nextbits ( ) != '0000 0000 0000 0000 0000 0001') {	
Sequence_extension_data	8
}	
Next_start_code( )	
}	
if (nextbits( ) == user_data_start_code) {	
User_data_start_code	32
While (nextbits( ) != '0000 0000 0000 0000 0000 0001') {	
User_data	8
}	
Next_start_code( )	
}	
}	

FIG.10

Group of pictures layer

SYNTAX	BIT NUMBER
Group_of_pictures( ) {	
Group_start_code	32
Time_code	25
Closed_gop	1
Broken_link	1
Next_start_code( )	
if (nextbits( ) == extension_start_code) {	
Extension_start_code	32
While (nextbits( ) != '0000 0000 0000 0000 0000 0001') {	
group_extension_data	8
}	
Next_start_code( )	
}	
if (nextbits( ) == user_data_start_code) {	
User_data_start_code	32
While (nextbits( ) != '0000 0000 0000 0000 0000 0001') {	
User_data	8
}	
Next_start_code( )	
}	
do {	
Picture( )	
} While (nextbits( ) == picture_start_code)	
}	

FIG.11

Picture layer

SYNTAX	BIT NUMBER
Picture( ) {	
picture_start_code	32
Temporal_reference	10
picture_coding_type	3
vbv_delay	16
if ((picture_coding_type = 2) | |	
(picture_coding_type = 3)) {	
full_pel_forward_vector	1
forward_f_code	3
}	
if (picture_coding_type = 3) {	
full_pel_backward_vector	1
backward_f_code	3
}	
while (nextbits( ) = '1') {	
extra_bit_picture	1
extra_information_picture	8
}	
extra_bit_picture	1
next_start_code( )	
if (nextbits( ) = extension_start_code) {	
extension_start_code	32
while (nextbits( ) != '0000 0000 0000 0000 0000 0001') {	
Picture_extension_data	8
}	
Next_start_code( )	
}	
if (nextbits( ) == user_data_start_code) {	
User_data_start_code	32
While (nextbits( ) != '0000 0000 0000 0000 0000 0001') {	
User data	8
}	
Next_start_code( )	
}	
do {	
Slice( )	
} while (nextbits( ) = slice_start_code)	
}	

FIG.12

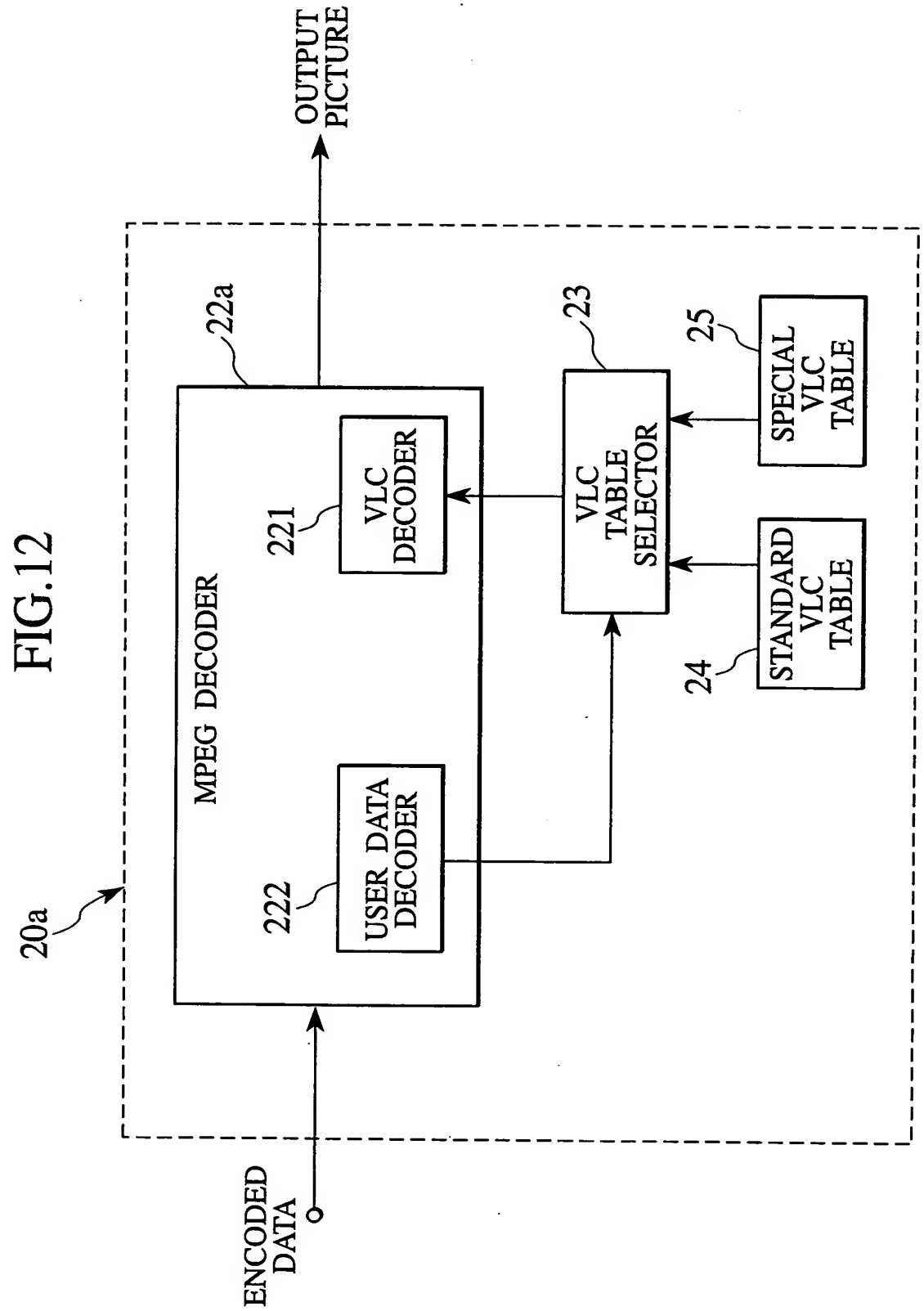

OUTPUT PICTURE
22a
MPEG DECODER
221
VLC DECODER
222
USER DATA DECODER
23
VLC TABLE SELECTOR
25
SPECIAL VLC TABLE
24
STANDARD VLC TABLE
20a
ENCODED DATA

FIG.13

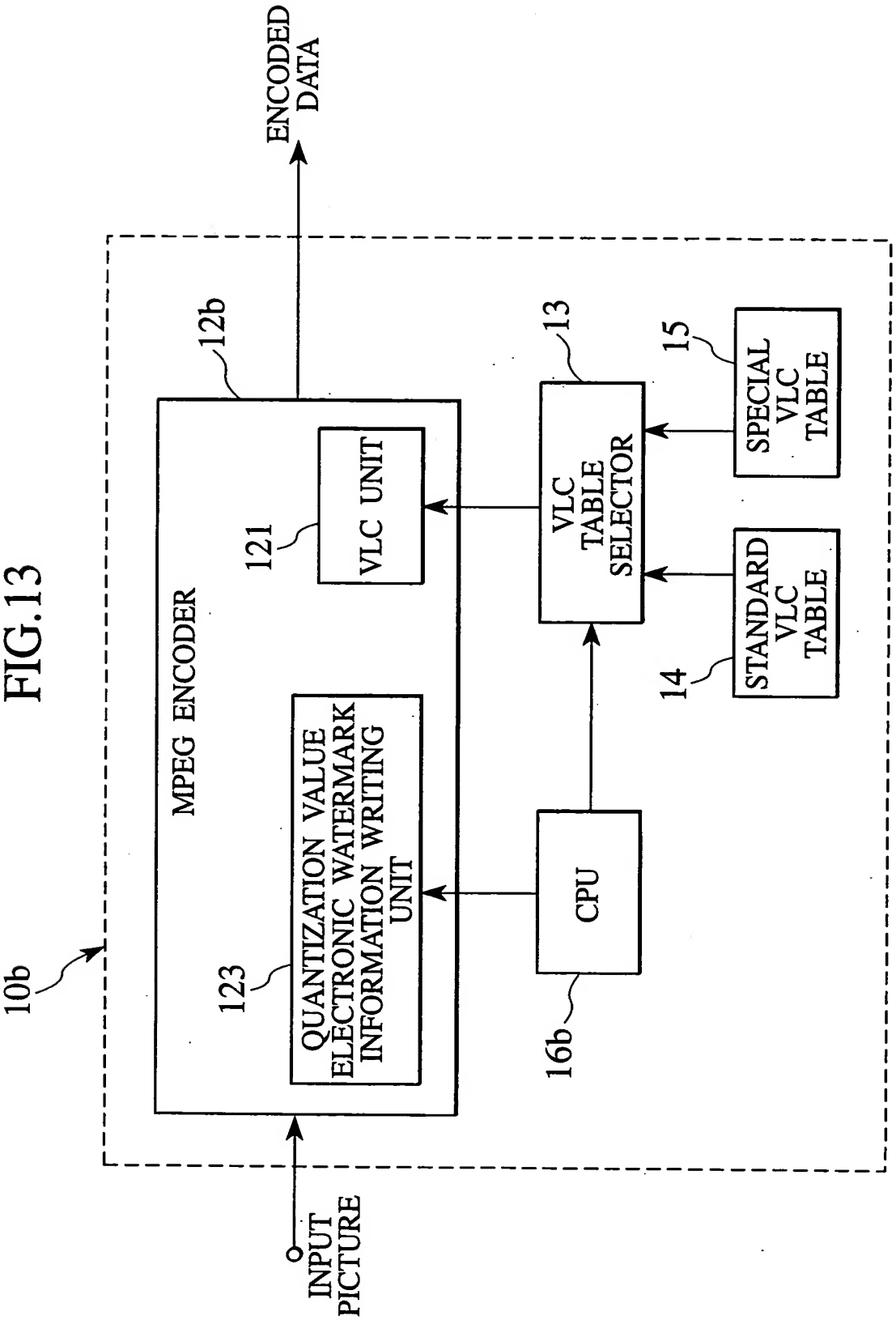

ENCODED DATA
12b
MPEG ENCODER
121
VLC UNIT
123
QUANTIZATION VALUE ELECTRONIC WATERMARK INFORMATION WRITING UNIT
13
VLC TABLE SELECTOR
15
SPECIAL VLC TABLE
14
STANDARD VLC TABLE
CPU
16b
10b
INPUT PICTURE

FIG.14

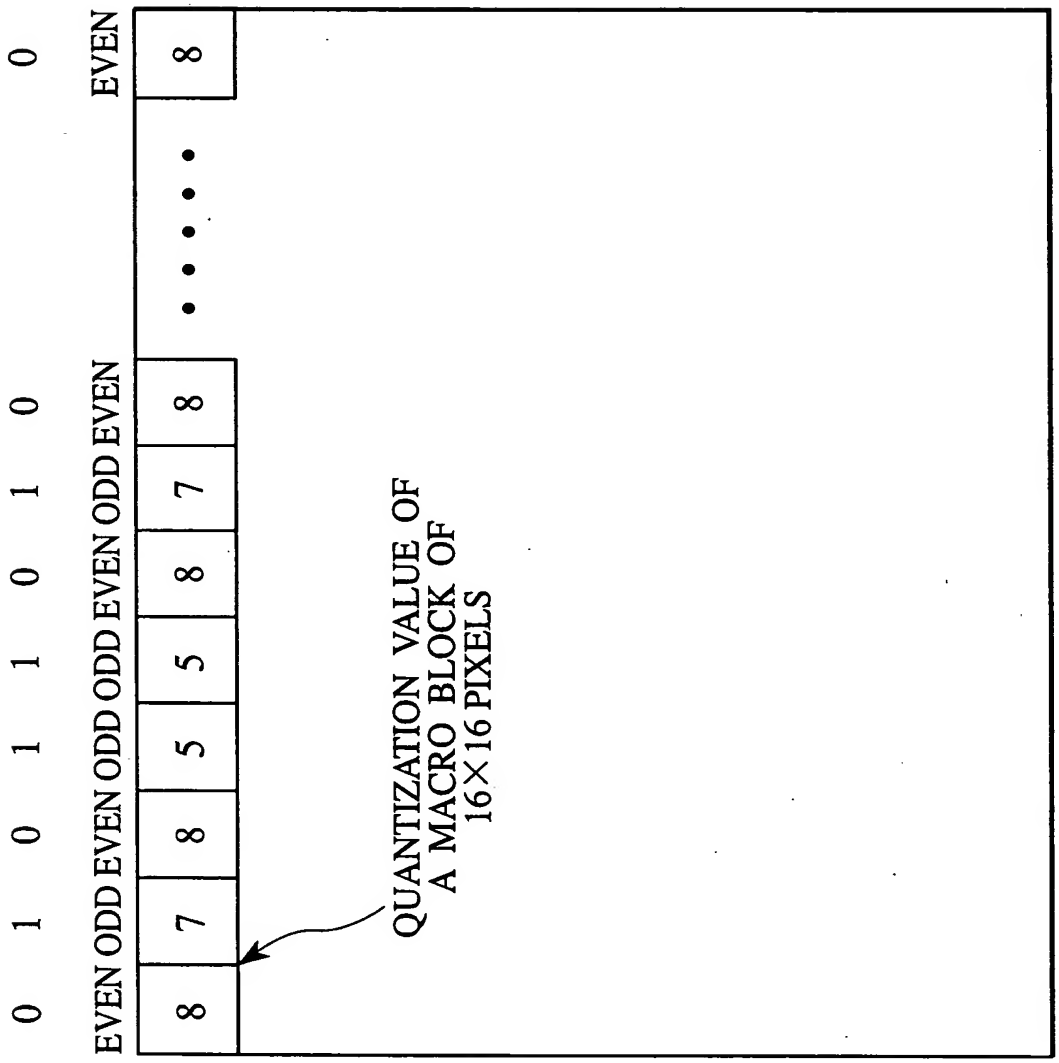

0 1 0 1 1 0 1 0 0
EVEN ODD EVEN ODD ODD EVEN ODD EVEN EVEN
8 7 8 5 5 8 7 8 8
QUANTIZATION VALUE OF A MACRO BLOCK OF 16×16 PIXELS

FIG.15

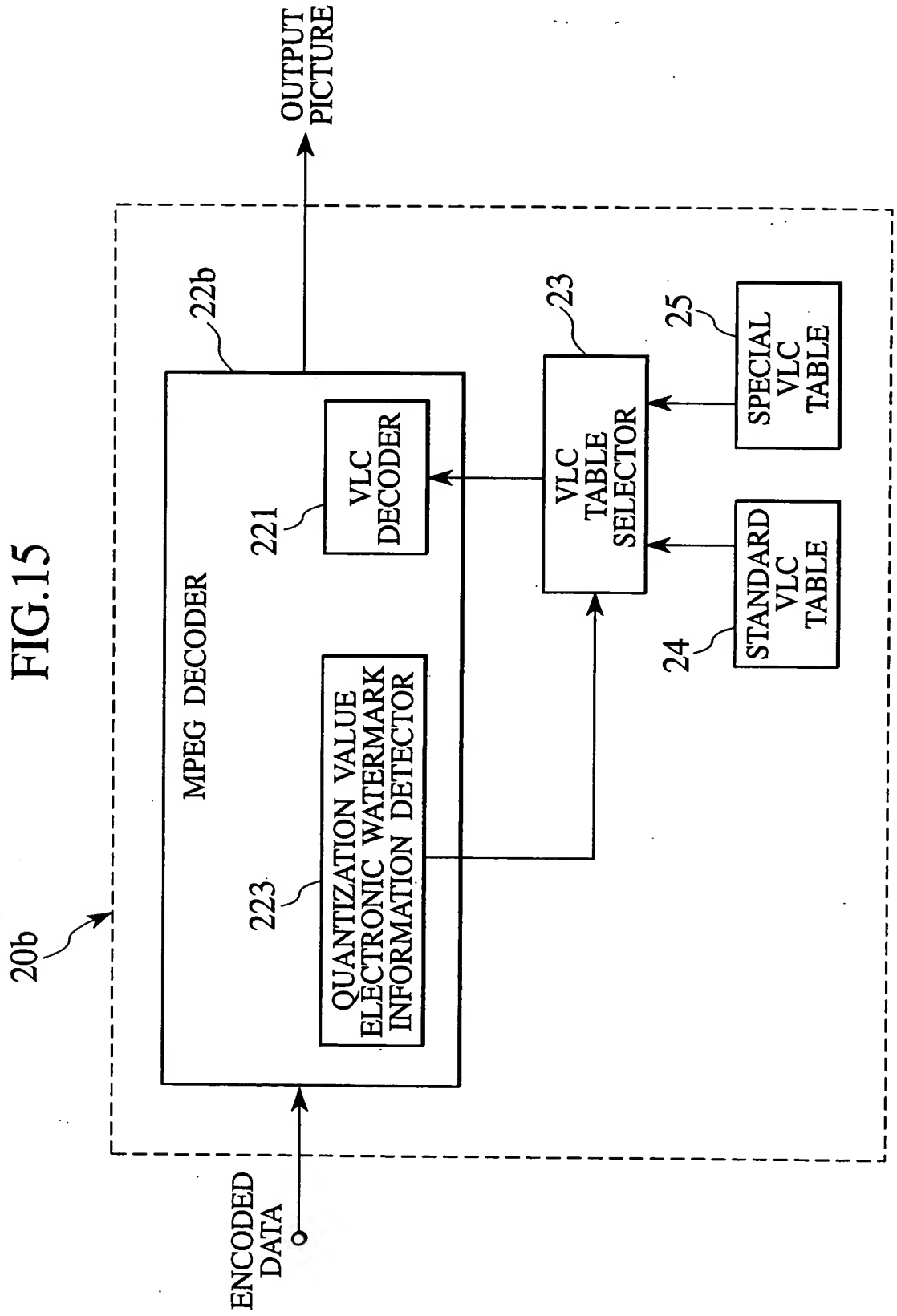

20b
ENCODED DATA
MPEG DECODER
22b
OUTPUT PICTURE
221
VLC DECODER
223
QUANTIZATION VALUE ELECTRONIC WATERMARK INFORMATION DETECTOR
23
VLC TABLE SELECTOR
24
STANDARD VLC TABLE
25
SPECIAL VLC TABLE

FIG.16

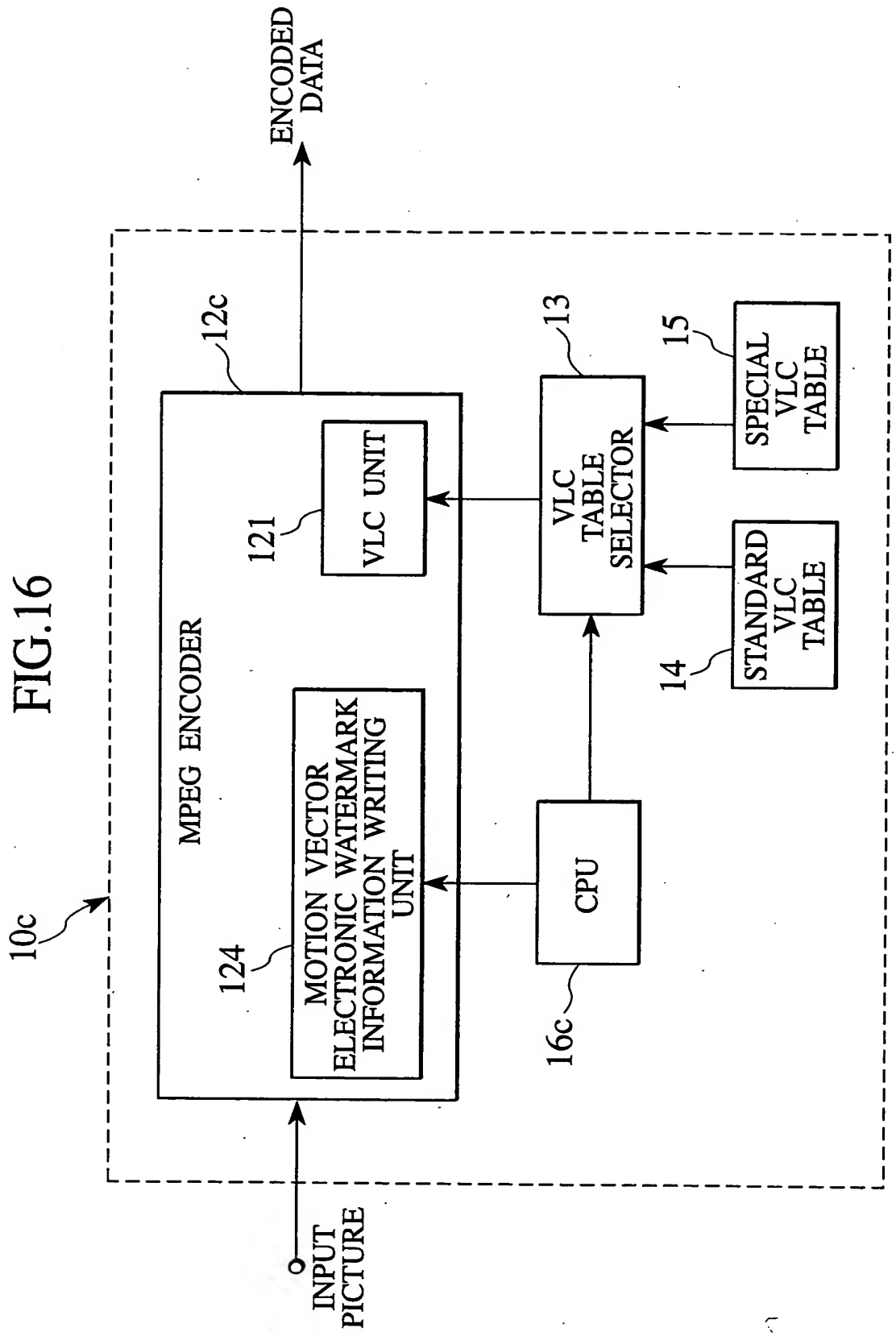

10c
INPUT PICTURE
MPEG ENCODER
12c
ENCODED DATA
121
VLC UNIT
124
MOTION VECTOR ELECTRONIC WATERMARK INFORMATION WRITING UNIT
CPU
16c
VLC TABLE SELECTOR
13
STANDARD VLC TABLE
14
SPECIAL VLC TABLE
15

FIG.17

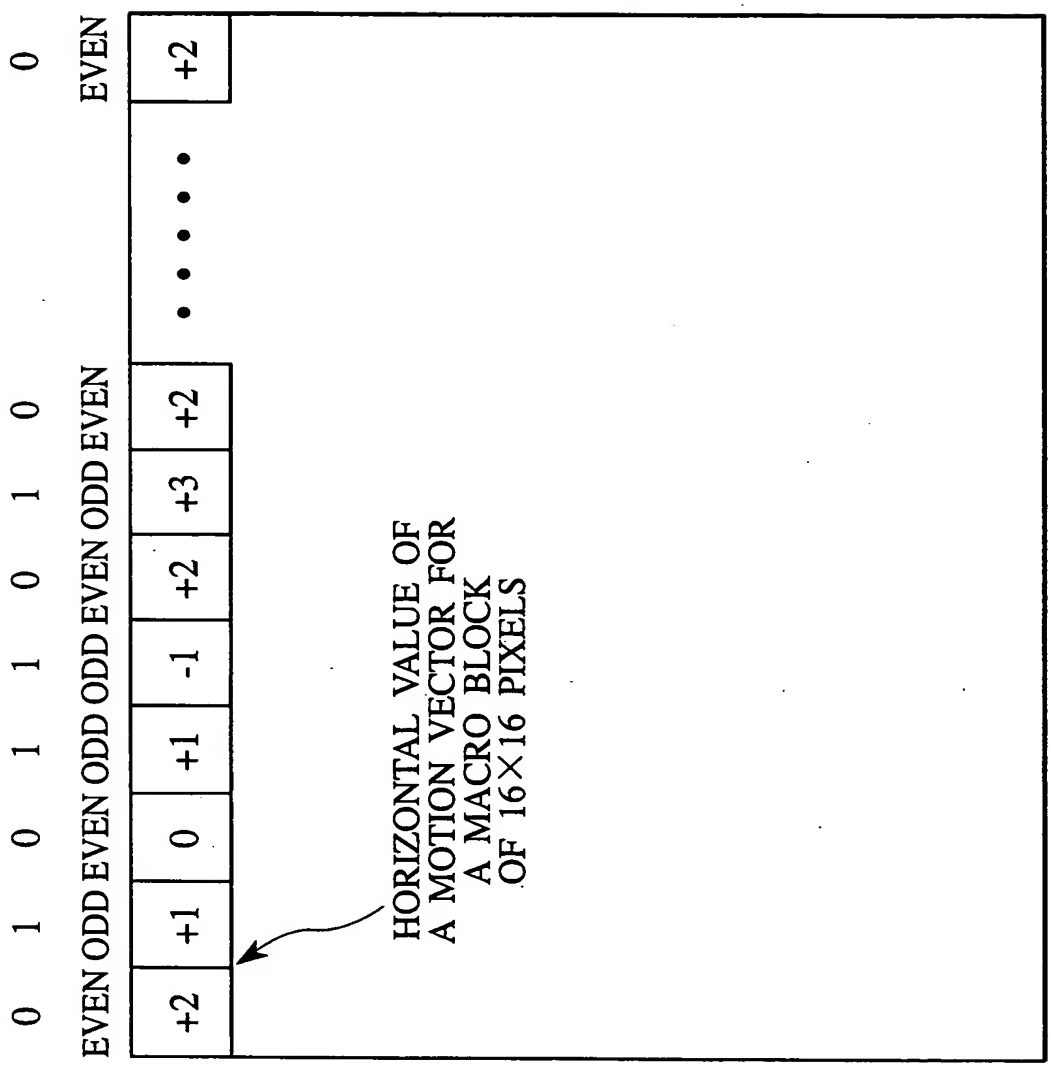

0 1 0 1 1 0 1 0 0
EVEN ODD EVEN ODD ODD EVEN ODD EVEN EVEN
+2 +1 0 +1 -1 +2 +3 +2 • • • • • +2
HORIZONTAL VALUE OF
A MOTION VECTOR FOR
A MACRO BLOCK
OF 16×16 PIXELS

FIG.18

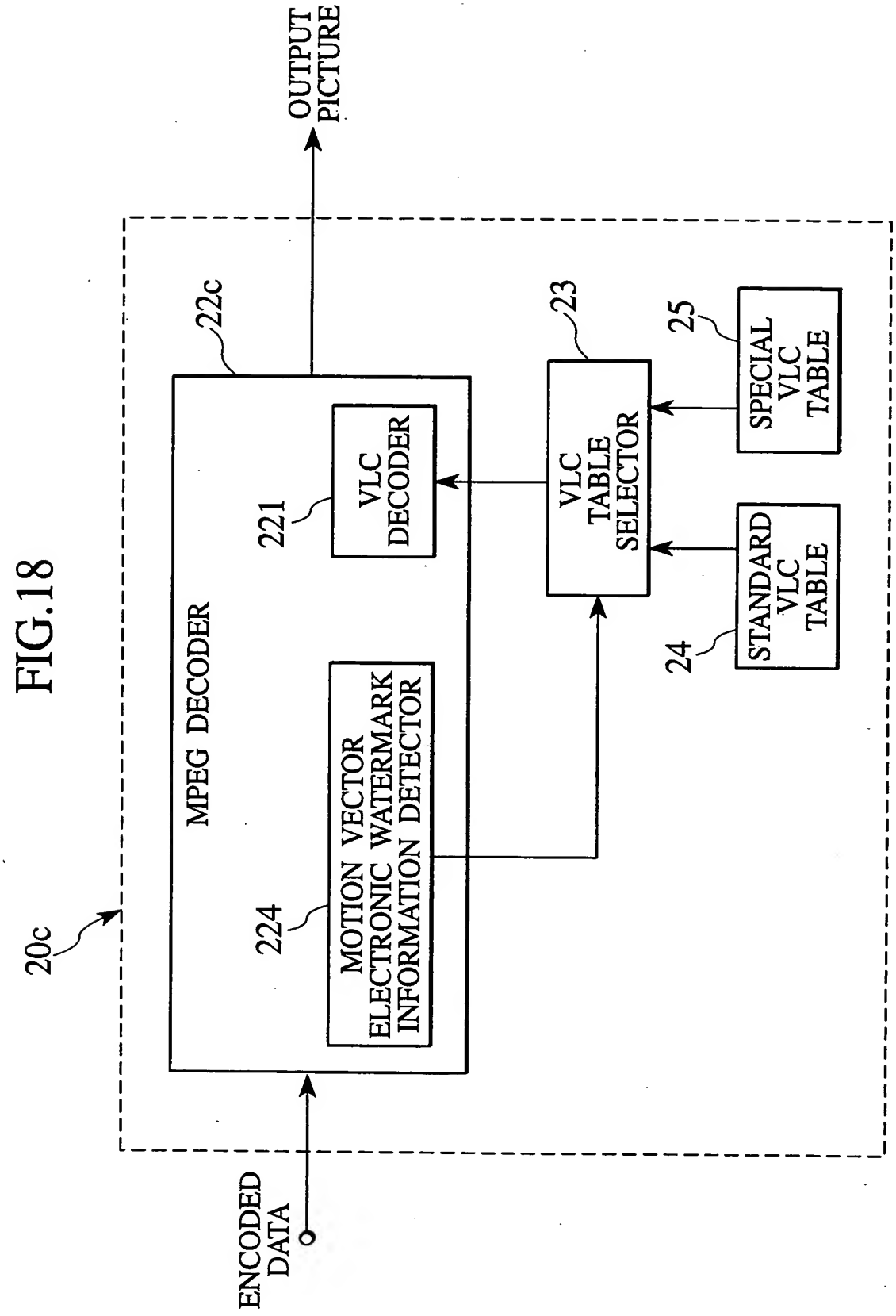

OUTPUT PICTURE
22c
23
25
SPECIAL VLC TABLE
MPEG DECODER
221
VLC DECODER
VLC TABLE SELECTOR
STANDARD VLC TABLE
24
224
MOTION VECTOR ELECTRONIC WATERMARK INFORMATION DETECTOR
20c
ENCODED DATA

FIG.19

VLC CODE	RUN LENGTH	LEVEL
10 (Note 2)	End of block	
1 s (Note 3)	0	1
11 s (Note 4)	0	1
011 s	1	1
0100 s	0	2
0101 s	2	1
0010 1 s	0	3
0011 1 s	3	1
0011 0 s	4	1
0001 10 s	1	2
0001 11 s	5	1
0001 01 s	6	1
0001 00 s	7	1
0000 110 s	0	4
0000 100 s	2	2
0000 111 s	8	1
0000 101 s	9	1
0000 01	Escape	
0010 0110 s	0	5
0010 0001 s	0	6
0010 0101 s	1	3
0010 0100 s	3	2
0010 0111 s	10	1
0010 0011 s	11	1
. . . . .	. .	.
. . . . .	. .	.
. . . . .	. .	.
0000 0000 0001 1111 s	27	1
0000 0000 0001 1110 s	28	1
0000 0000 0001 1101 s	29	1
0000 0000 0001 1100 s	30	1
0000 0000 0001 1011 s	31	1
(NOTE) AS FOR BIT "s", "0" REPRESENTS POSITIVE AND "1" REPRESENTS NEGATIVE.		

FIG.20

VLC CODE	RUN LENGTH	LEVEL	REPLACED ADDRESS
10 (Note 2)	End of block		
1 s (Note 3)	0	1	A1
11 s (Note 4)	0	1	A2
011 s	1	1	B1
0100 s	0	2	A3
0101 s	2	1	C1
0010 1 s	0	3	A4
0011 1 s	3	1	
0011 0 s	4	1	
0001 10 s	1	2	B2
0001 11 s	5	1	
0001 01 s	6	1	
0001 00 s	7	1	
0000 110 s	0	4	A5
0000 100 s	2	2	C2
0000 111 s	8	1	
0000 101 s	9	1	
0000 01	Escape		
0010 0110 s	0	5	A6
0010 0001 s	0	6	A7
0010 0101 s	1	3	B3
. . . . .	.	.	
. . . . .	.	.	
0000 0010 00 s	16	1	
0000 0001 1101 s	0	8	A8
0000 0001 1000 s	0	9	A9
0000 0001 0011 s	0	10	A10
0000 0001 0000 s	0	11	A11
0000 0001 1011 s	1	5	B4
. . . . .	. .	.	
. . . . .	. .	.	
0000 0001 0110 s	21	1	
0000 0000 1101 0 s	0	12	A12
0000 0000 1100 1 s	0	13	A13
0000 0000 1100 0 s	0	14	A14
0000 0000 1011 1 s	0	15	A15
0000 0000 1011 0 s	1	6	B5
. . . . .	.	.	

FIG.21

VLC CODE	RUN LENGTH	LEVEL	REPLACED ADDRESS
. . . . .	.	. .	
. . . . .	.	. .	
0000 0000 1110 0 s	25	1	
0000 0000 1101 1 s	26	1	
0000 0000 0111 11 s	0	16	
0000 0000 0111 10 s	0	17	A16
0000 0000 0111 01 s	0	18	A17
0000 0000 0111 00 s	0	19	A18
0000 0000 0110 11 s	0	20	A19
0000 0000 0110 10 s	0	21	A20
0000 0000 0110 01 s	0	22	A21
0000 0000 0110 00 s	0	23	A22
0000 0000 0101 11 s	0	24	A23
0000 0000 0101 10 s	0	25	A24
0000 0000 0101 01 s	0	26	A25
0000 0000 0101 00 s	0	27	A26
0000 0000 0100 11 s	0	28	A27
0000 0000 0100 10 s	0	29	A28
0000 0000 0100 01 s	0	30	A29
0000 0000 0100 00 s	0	31	A30
0000 0000 0011 000 s	0	32	A31
0000 0000 0010 111 s	0	33	A32
0000 0000 0010 110 s	0	34	A33
0000 0000 0010 101 s	0	35	A34
0000 0000 0010 100 s	0	38	A35
0000 0000 0010 011 s	0	39	A36
0000 0000 0010 010 s	0	40	A37
0000 0000 0010 001 s	0	41	A38
. . . . .	.	. .	
. . . . .	.	. .	
0000 0000 0001 1100 s	30	1	
0000 0000 0001 1011 s	31	1	

FIG.22

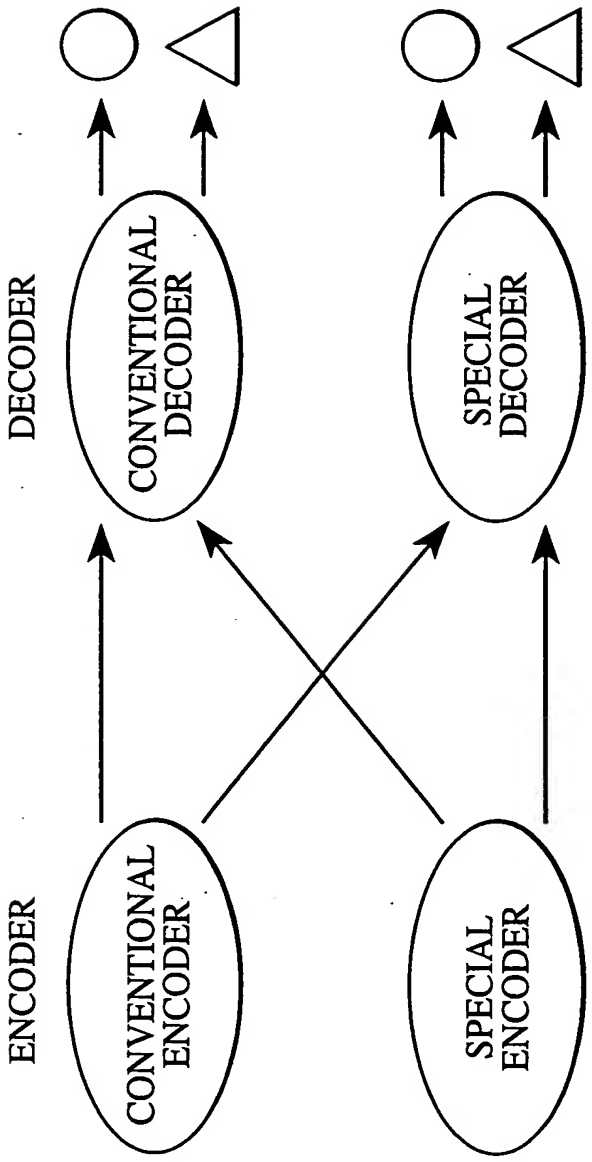

ENCODER
DECODER
CONVENTIONAL ENCODER
SPECIAL ENCODER
CONVENTIONAL DECODER
SPECIAL DECODER

FIG.23

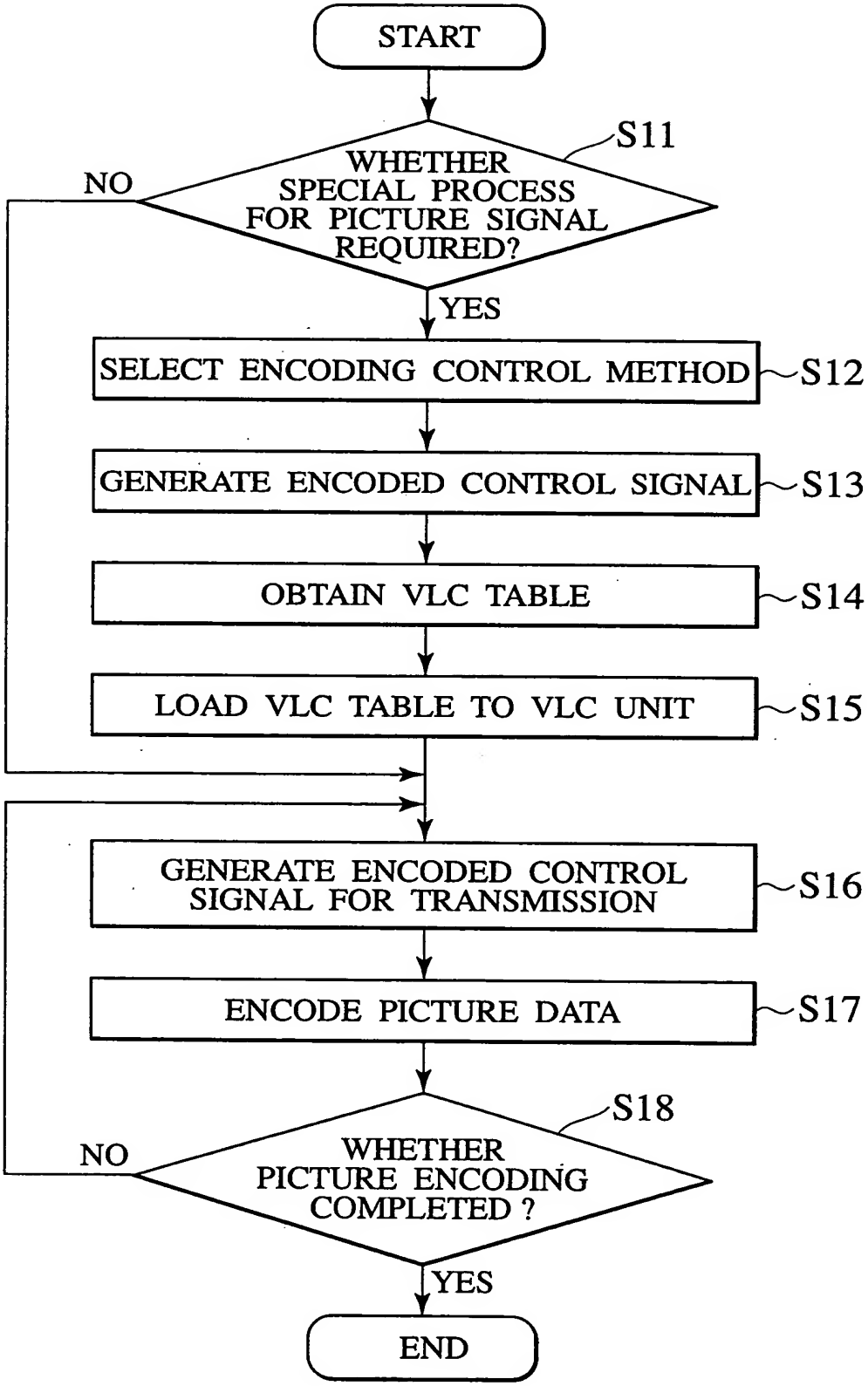

START
S11
WHETHER SPECIAL PROCESS FOR PICTURE SIGNAL REQUIRED?
NO
YES
SELECT ENCODING CONTROL METHOD
S12
GENERATE ENCODED CONTROL SIGNAL
S13
OBTAIN VLC TABLE
S14
LOAD VLC TABLE TO VLC UNIT
S15
GENERATE ENCODED CONTROL SIGNAL FOR TRANSMISSION
S16
ENCODE PICTURE DATA
S17
S18
WHETHER PICTURE ENCODING COMPLETED ?
NO
YES
END

FIG.24

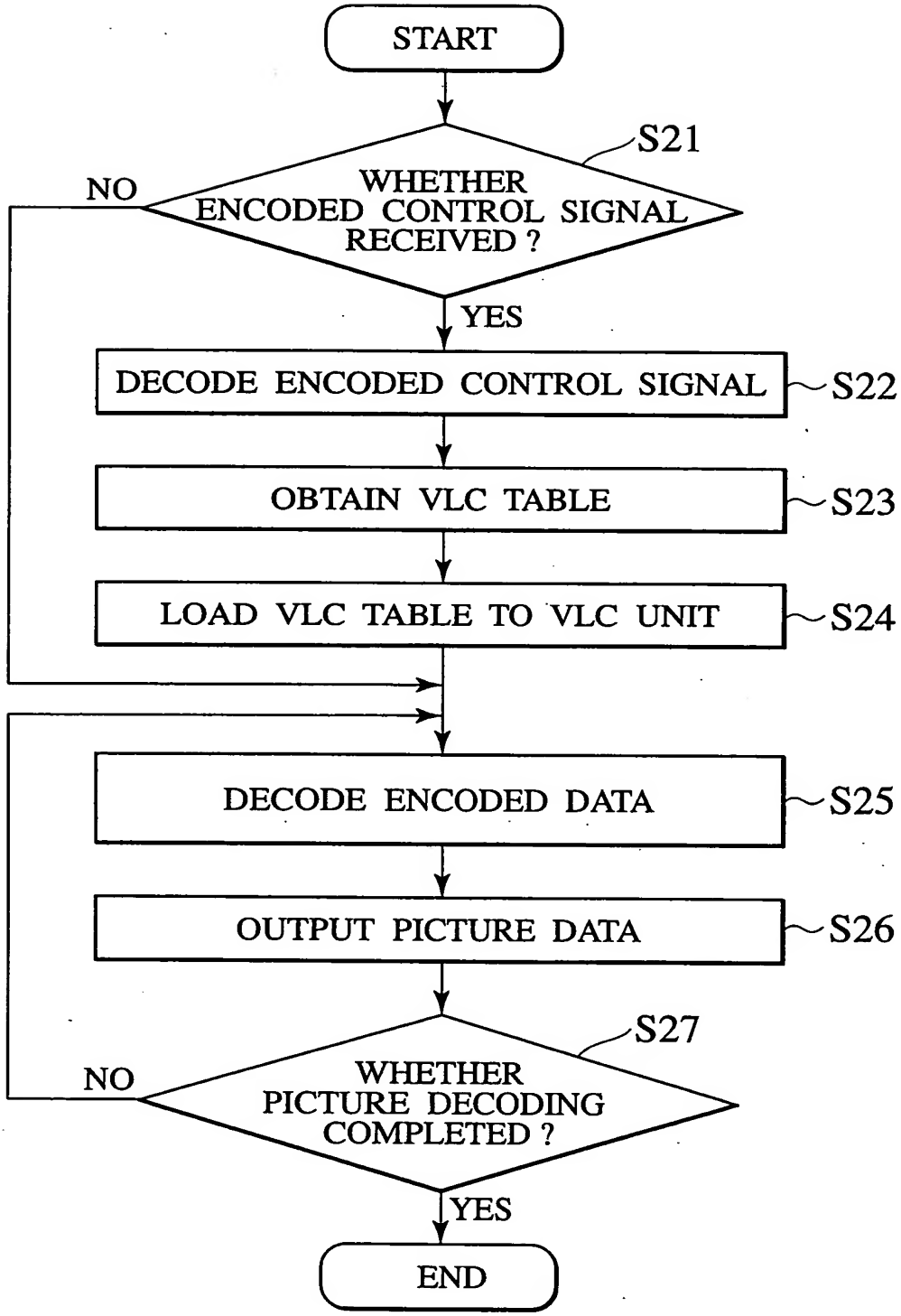

START
S21
WHETHER ENCODED CONTROL SIGNAL RECEIVED ?
NO
YES
DECODE ENCODED CONTROL SIGNAL
S22
OBTAIN VLC TABLE
S23
LOAD VLC TABLE TO VLC UNIT
S24
DECODE ENCODED DATA
S25
OUTPUT PICTURE DATA
S26
S27
WHETHER PICTURE DECODING COMPLETED ?
NO
YES
END

FIG.25

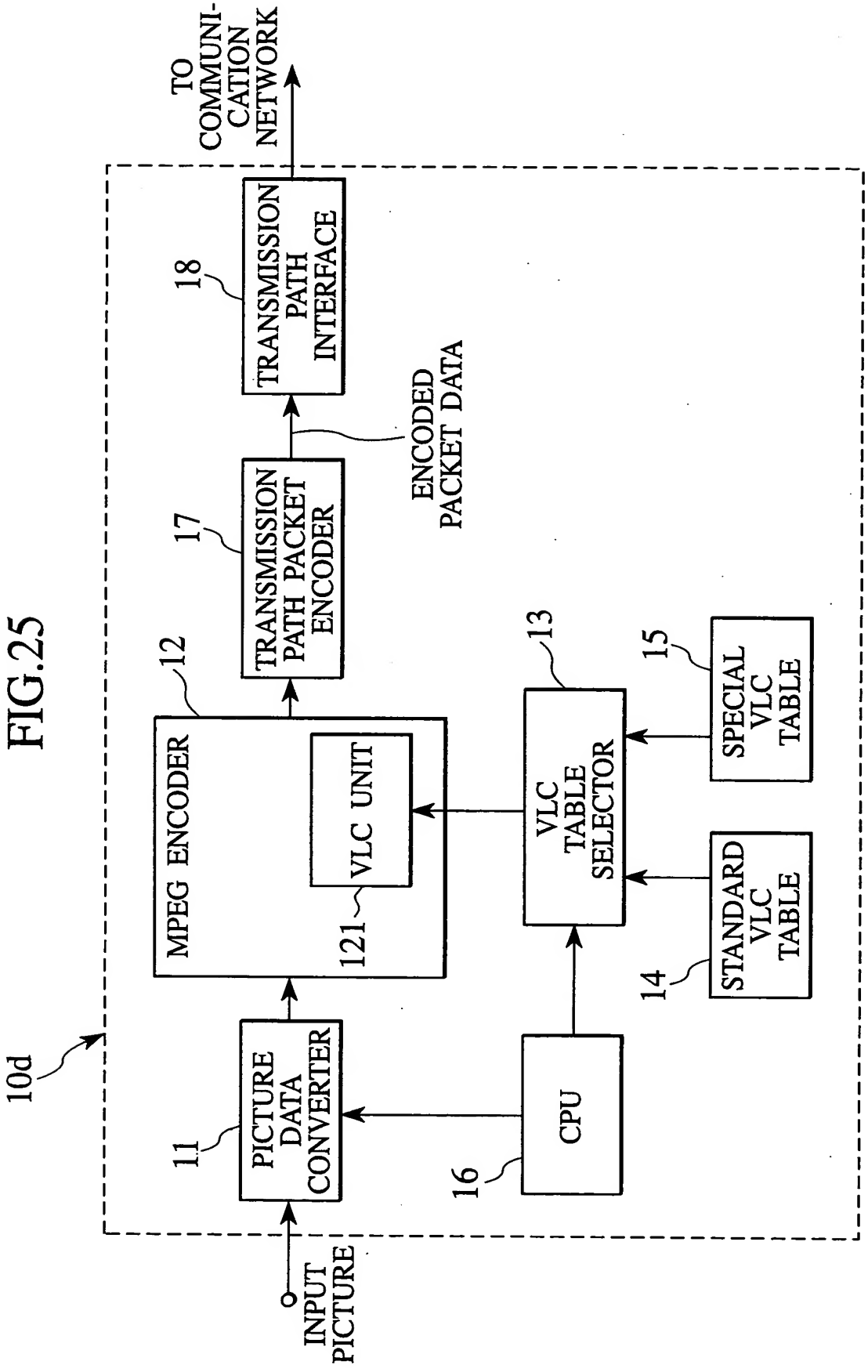

10d
INPUT PICTURE
11
PICTURE DATA CONVERTER
16
CPU
MPEG ENCODER
12
VLC UNIT
121
VLC TABLE SELECTOR
13
STANDARD VLC TABLE
14
SPECIAL VLC TABLE
15
TRANSMISSION PATH PACKET ENCODER
17
ENCODED PACKET DATA
TRANSMISSION PATH INTERFACE
18
TO COMMUNI-CATION NETWORK

FIG.26

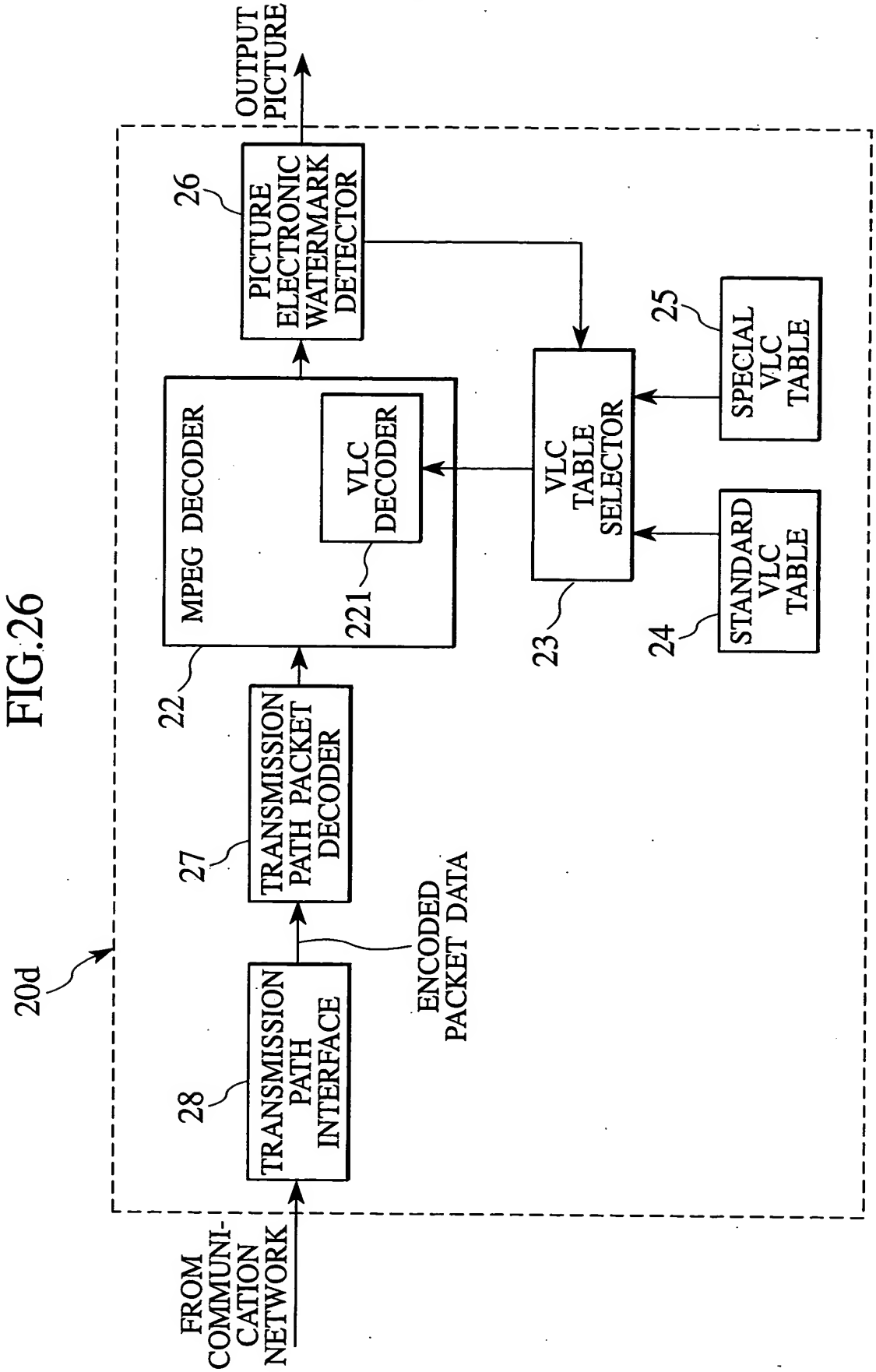

20d
FROM COMMUNI-CATION NETWORK
28
TRANSMISSION PATH INTERFACE
ENCODED PACKET DATA
27
TRANSMISSION PATH PACKET DECODER
22
MPEG DECODER
221
VLC DECODER
23
VLC TABLE SELECTOR
24
STANDARD VLC TABLE
25
SPECIAL VLC TABLE
26
PICTURE ELECTRONIC WATERMARK DETECTOR
OUTPUT PICTURE

FIG.27

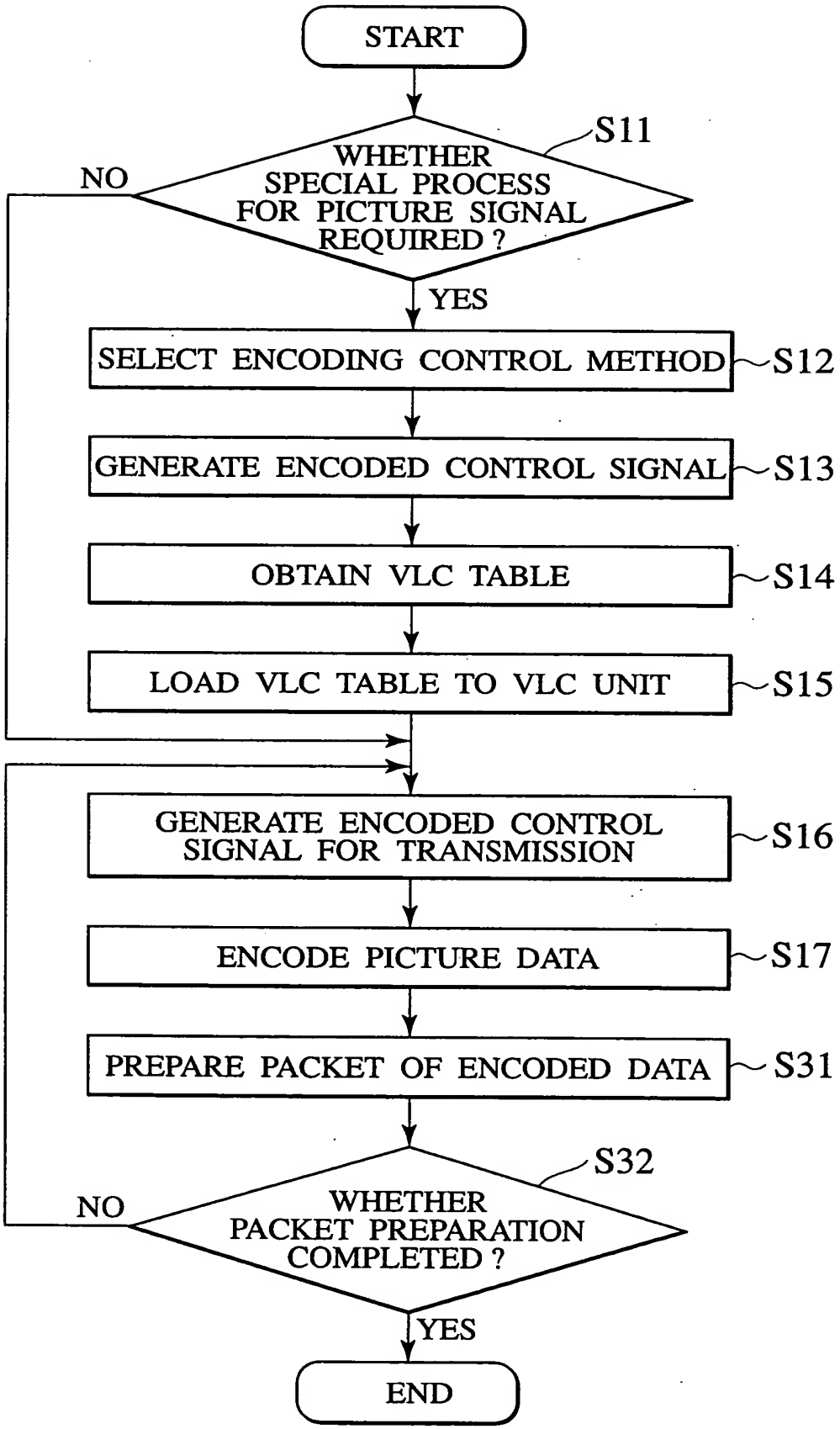

START
S11
WHETHER SPECIAL PROCESS FOR PICTURE SIGNAL REQUIRED ?
NO
YES
SELECT ENCODING CONTROL METHOD
S12
GENERATE ENCODED CONTROL SIGNAL
S13
OBTAIN VLC TABLE
S14
LOAD VLC TABLE TO VLC UNIT
S15
GENERATE ENCODED CONTROL SIGNAL FOR TRANSMISSION
S16
ENCODE PICTURE DATA
S17
PREPARE PACKET OF ENCODED DATA
S31
S32
WHETHER PACKET PREPARATION COMPLETED ?
NO
YES
END

FIG.28

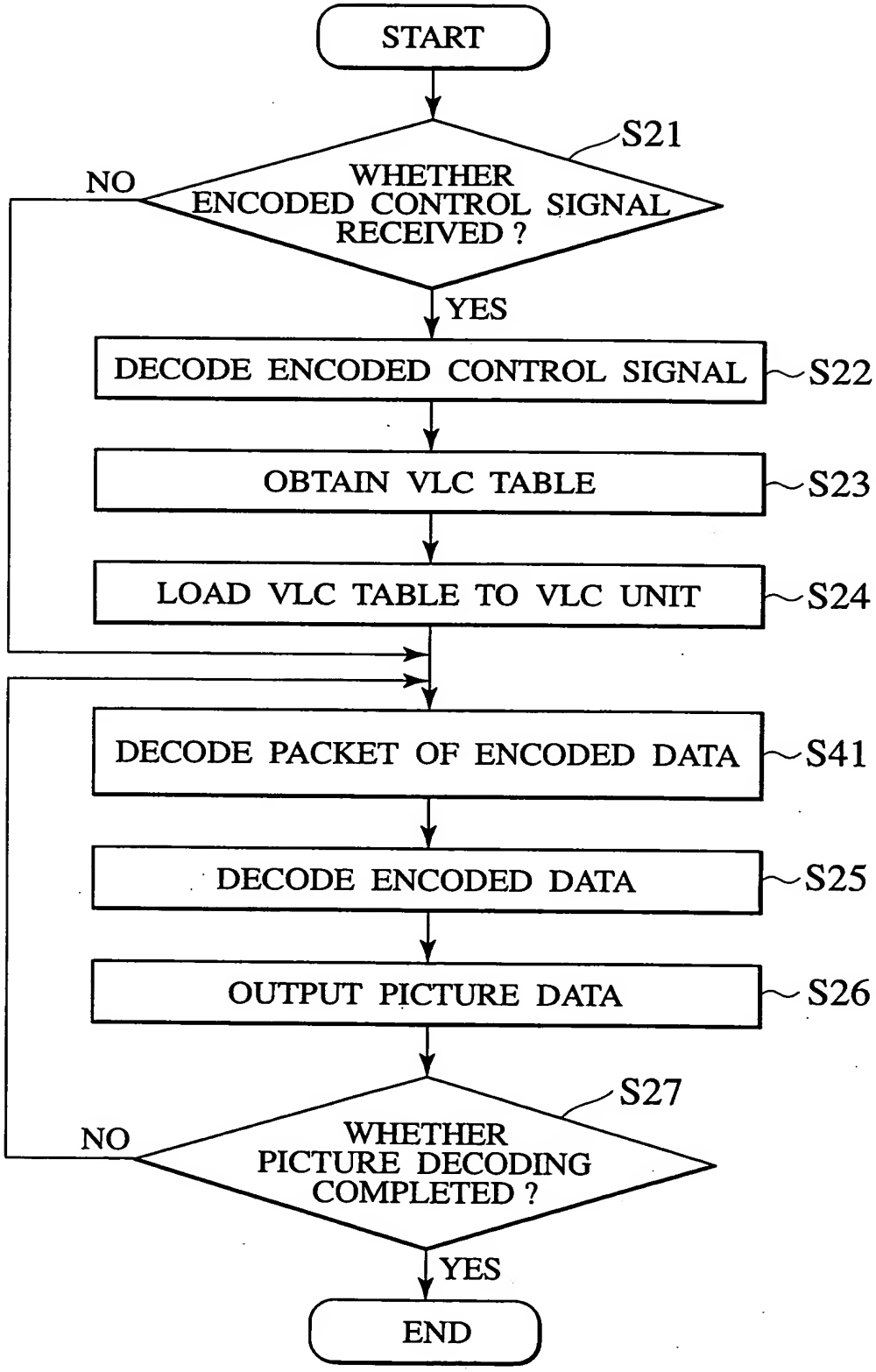

START
S21
WHETHER ENCODED CONTROL SIGNAL RECEIVED ?
NO
YES
DECODE ENCODED CONTROL SIGNAL
S22
OBTAIN VLC TABLE
S23
LOAD VLC TABLE TO VLC UNIT
S24
DECODE PACKET OF ENCODED DATA
S41
DECODE ENCODED DATA
S25
OUTPUT PICTURE DATA
S26
S27
WHETHER PICTURE DECODING COMPLETED ?
NO
YES
END

FIG.29

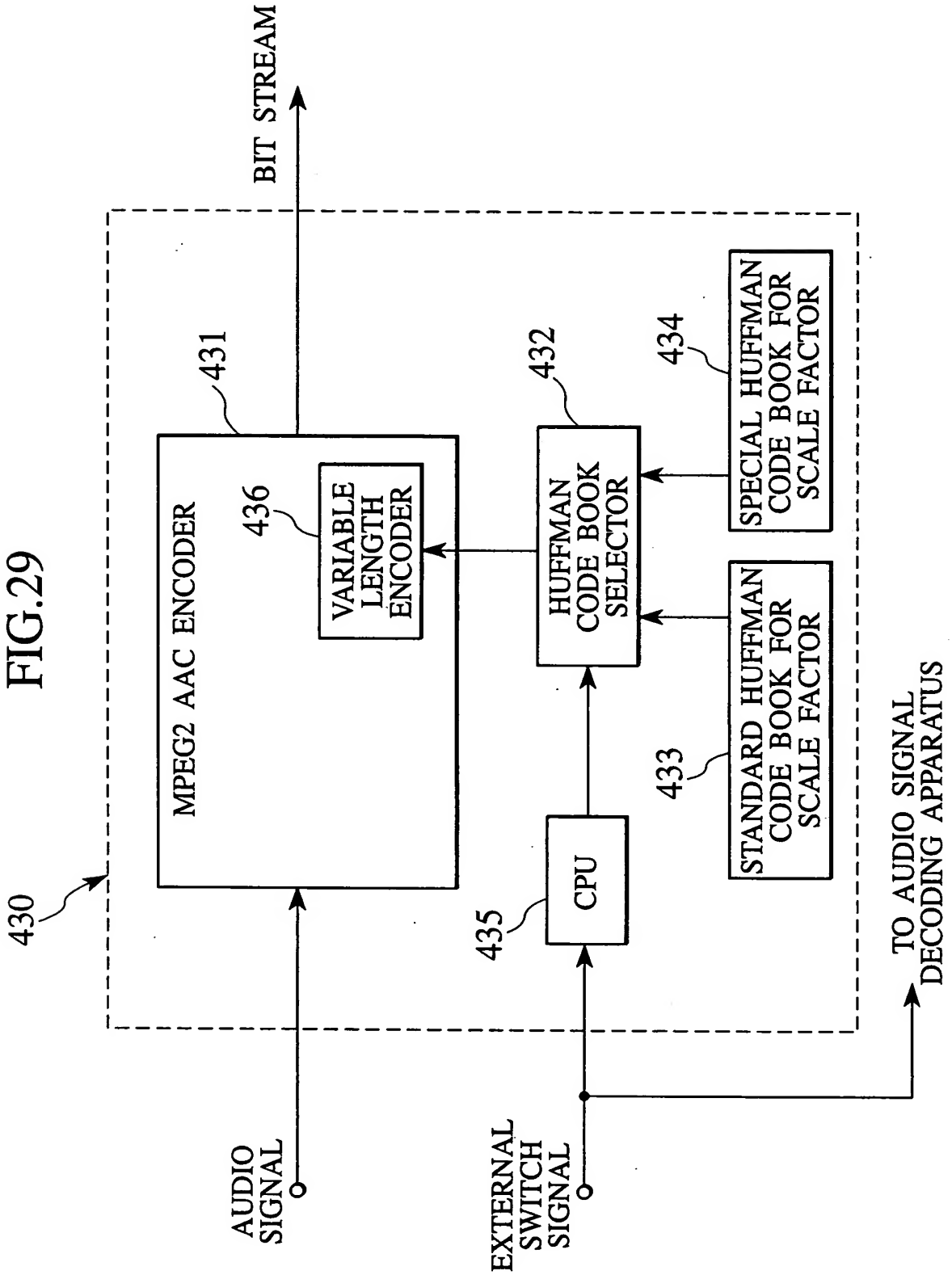

430
MPEG2 AAC ENCODER
431
BIT STREAM
436
VARIABLE LENGTH ENCODER
HUFFMAN CODE BOOK SELECTOR
432
SPECIAL HUFFMAN CODE BOOK FOR SCALE FACTOR
434
STANDARD HUFFMAN CODE BOOK FOR SCALE FACTOR
433
CPU
435
AUDIO SIGNAL
EXTERNAL SWITCH SIGNAL
TO AUDIO SIGNAL DECODING APPARATUS

FIG.30

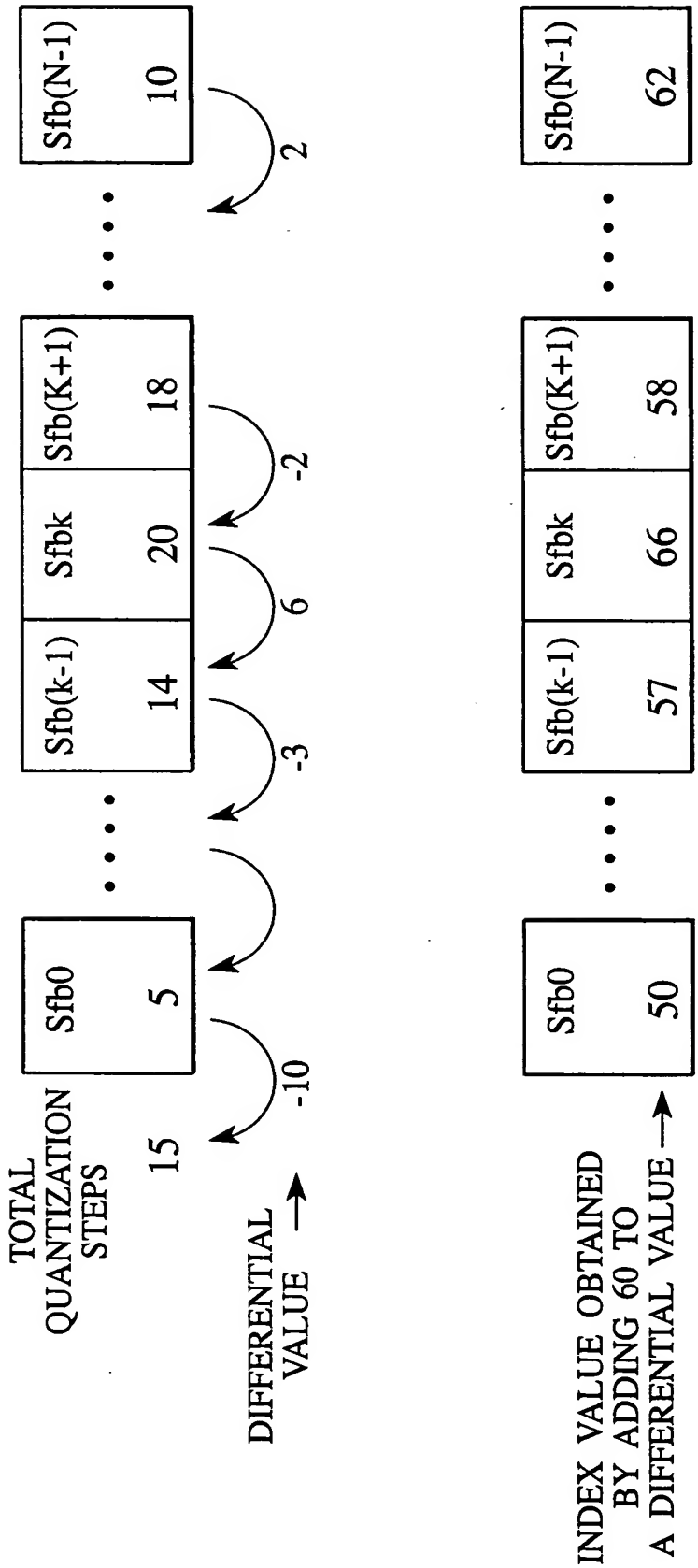

TOTAL QUANTIZATION STEPS
15
DIFFERENTIAL VALUE
-10
-3
6
-2
2
Sfb0
5
Sfb(k-1)
14
Sfbk
20
Sfb(K+1)
18
Sfb(N-1)
10
INDEX VALUE OBTAINED BY ADDING 60 TO A DIFFERENTIAL VALUE
Sfb0
50
Sfb(k-1)
57
Sfbk
66
Sfb(K+1)
58
Sfb(N-1)
62

FIG.31

index	length	codeword (HEXADECIMAL EXPRESSION)	index	length	codeword (HEXADECIMAL EXPRESSION)
0	18	3ffe8	61	4	a
:			62	4	c
10	19	7fff0	63	5	1b
:			64	6	39
20	19	7fffa	65	6	3b
:			:		
30	15	7ff6	70	9	1f6
:			:		
40	12	ff9	80	12	ff8
:			:		
50	9	1f7	90	18	3ffe2
:			:		
55	6	3a	100	19	7ffd4
56	6	38	:		
57	5	1a	110	19	7ffe6
58	4	b	:		
59	3	4	120	19	7fff3
60	1	0			

FIG.32

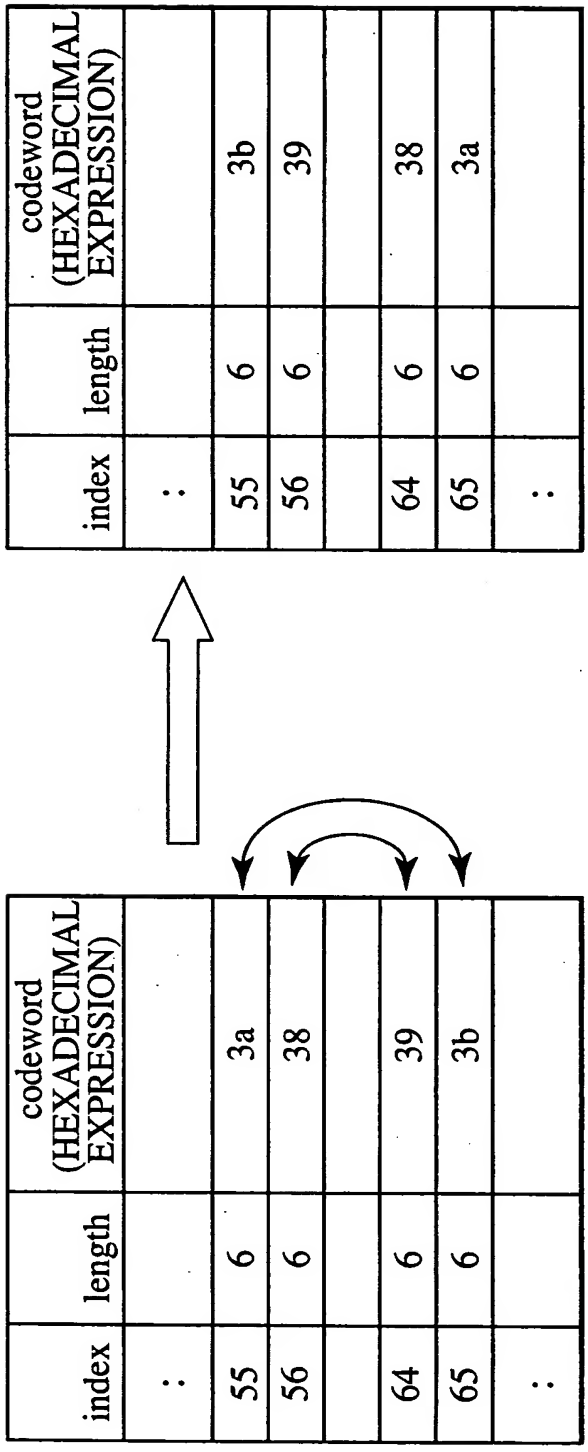

index
length
codeword (HEXADECIMAL EXPRESSION)
:
55
6
3a
56
6
38
64
6
39
65
6
3b
:
index
length
codeword (HEXADECIMAL EXPRESSION)
:
55
6
3b
56
6
39
64
6
38
65
6
3a
:

FIG.33

	sfb0	sfb1	sfb2	sfb3	sfb4
(1) SCALE FACTOR	10	15	19	14	10
(2) DIFFERENCE FROM PRECEDING SFB	-20	5	4	-5	-4
(3) OFFSET ADDITION	40	65	64	55	56
(4) CODEWORD ENCODING	ff9	3a	38	3b	39
(5) CODEWORD DECODING	40	55	56	65	64
(6) OFFSET SUBTRACTION	-20	-5	-4	5	4
(7) SCALE FACTOR	10	5	1	6	10

FIG.34

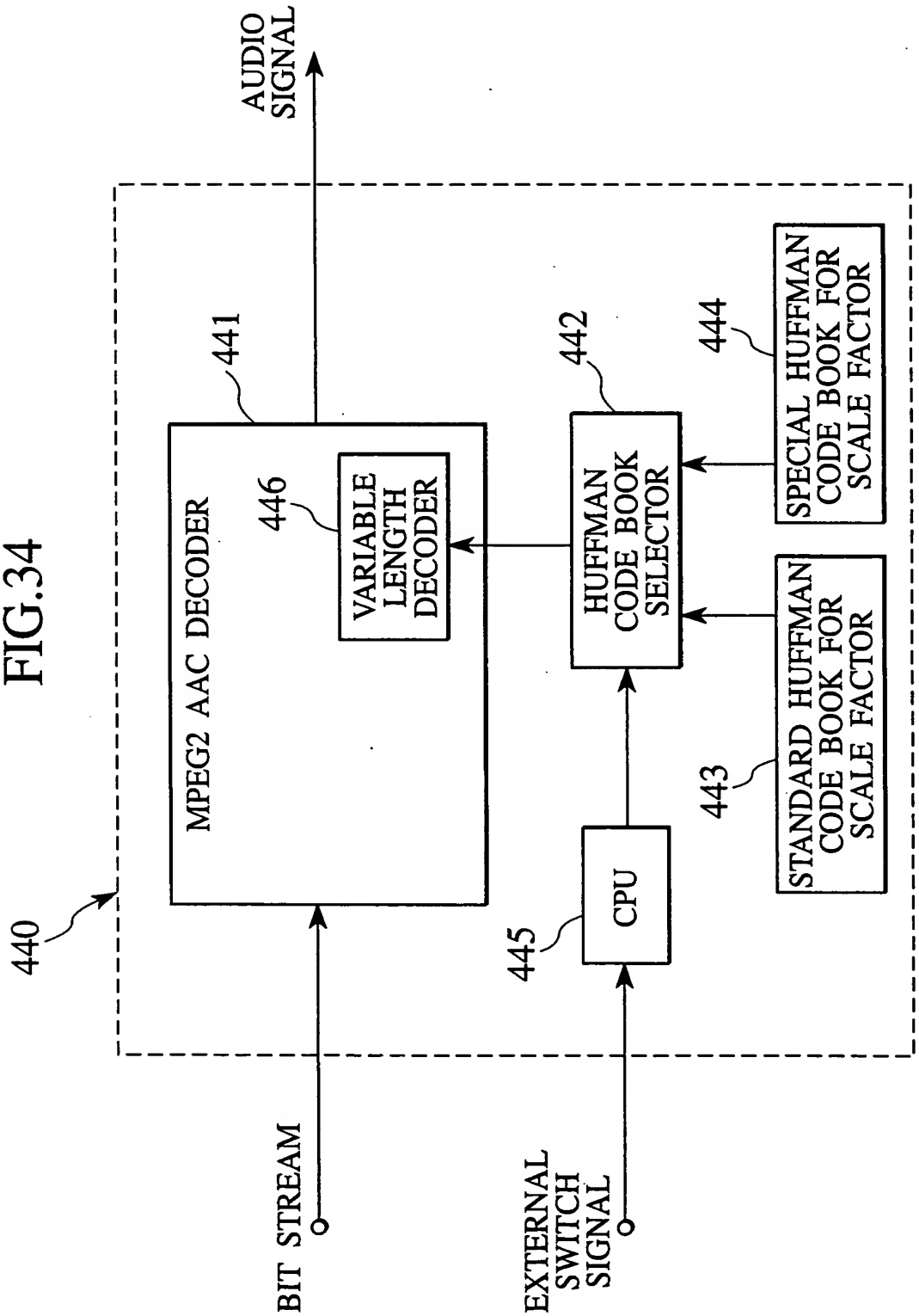

AUDIO SIGNAL
440
MPEG2 AAC DECODER
441
VARIABLE LENGTH DECODER
446
HUFFMAN CODE BOOK SELECTOR
442
SPECIAL HUFFMAN CODE BOOK FOR SCALE FACTOR
444
STANDARD HUFFMAN CODE BOOK FOR SCALE FACTOR
443
CPU
445
BIT STREAM
EXTERNAL SWITCH SIGNAL

FIG.35

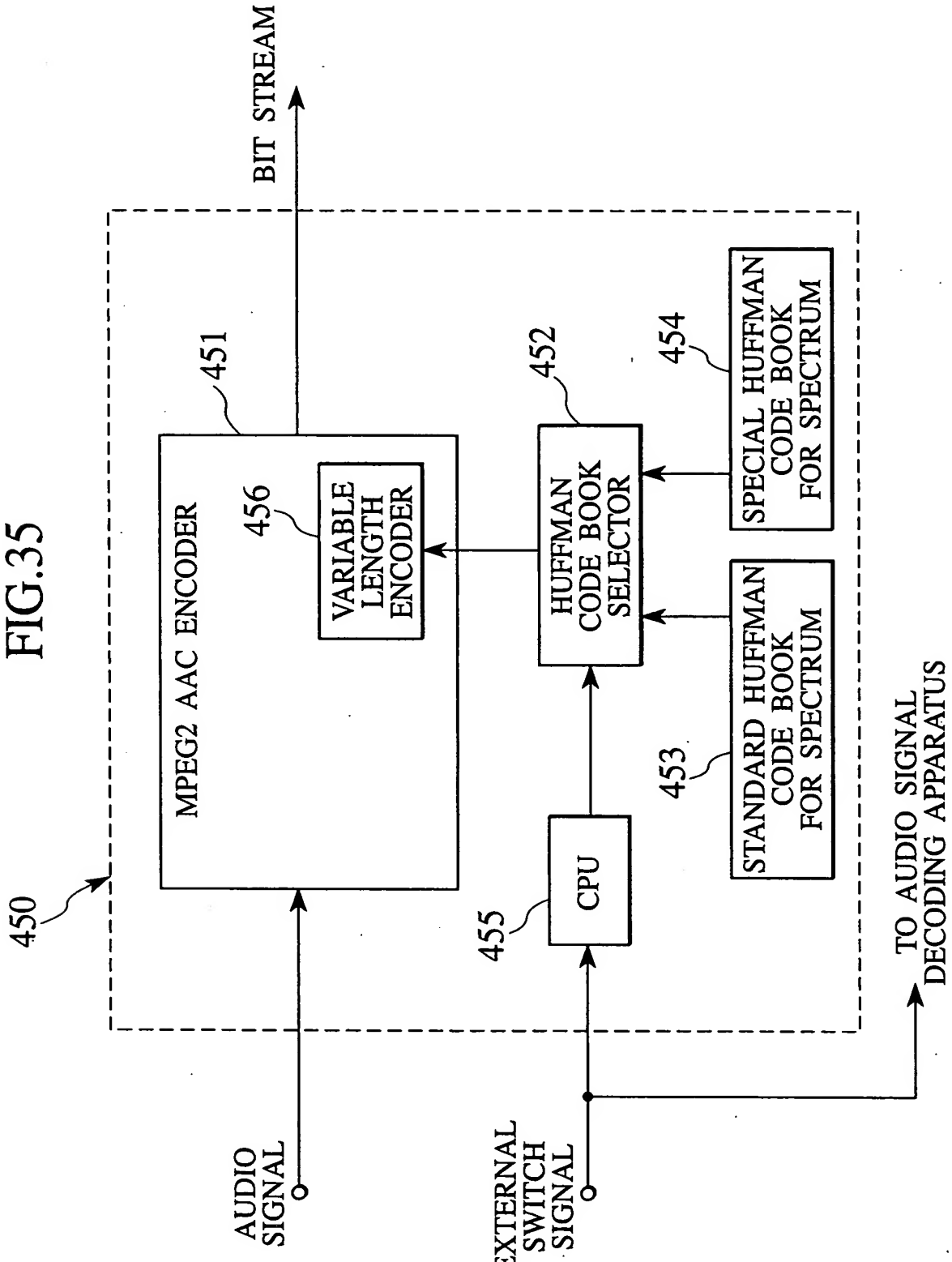

450
MPEG2 AAC ENCODER
451
BIT STREAM
456
VARIABLE LENGTH ENCODER
452
HUFFMAN CODE BOOK SELECTOR
454
SPECIAL HUFFMAN CODE BOOK FOR SPECTRUM
453
STANDARD HUFFMAN CODE BOOK FOR SPECTRUM
455
CPU
AUDIO SIGNAL
EXTERNAL SWITCH SIGNAL
TO AUDIO SIGNAL DECODING APPARATUS

FIG.36

index	length	codeword (HEXADECIMAL EXPRESSION)	index	length	codeword (HEXADECIMAL EXPRESSION)
0	9	1f3	41	5	7
:			42	6	1d
10	6	2d	43	5	b
:			44	6	30
20	8	f8	45	8	ef
:			:		
30	6	2c	50	6	29
:			:		
35	8	f5	60	8	f7
36	6	24	:		
37	5	8	70	6	25
38	6	1f	:		
39	5	9	80	9	1f6
40	3	0			

FIG.37

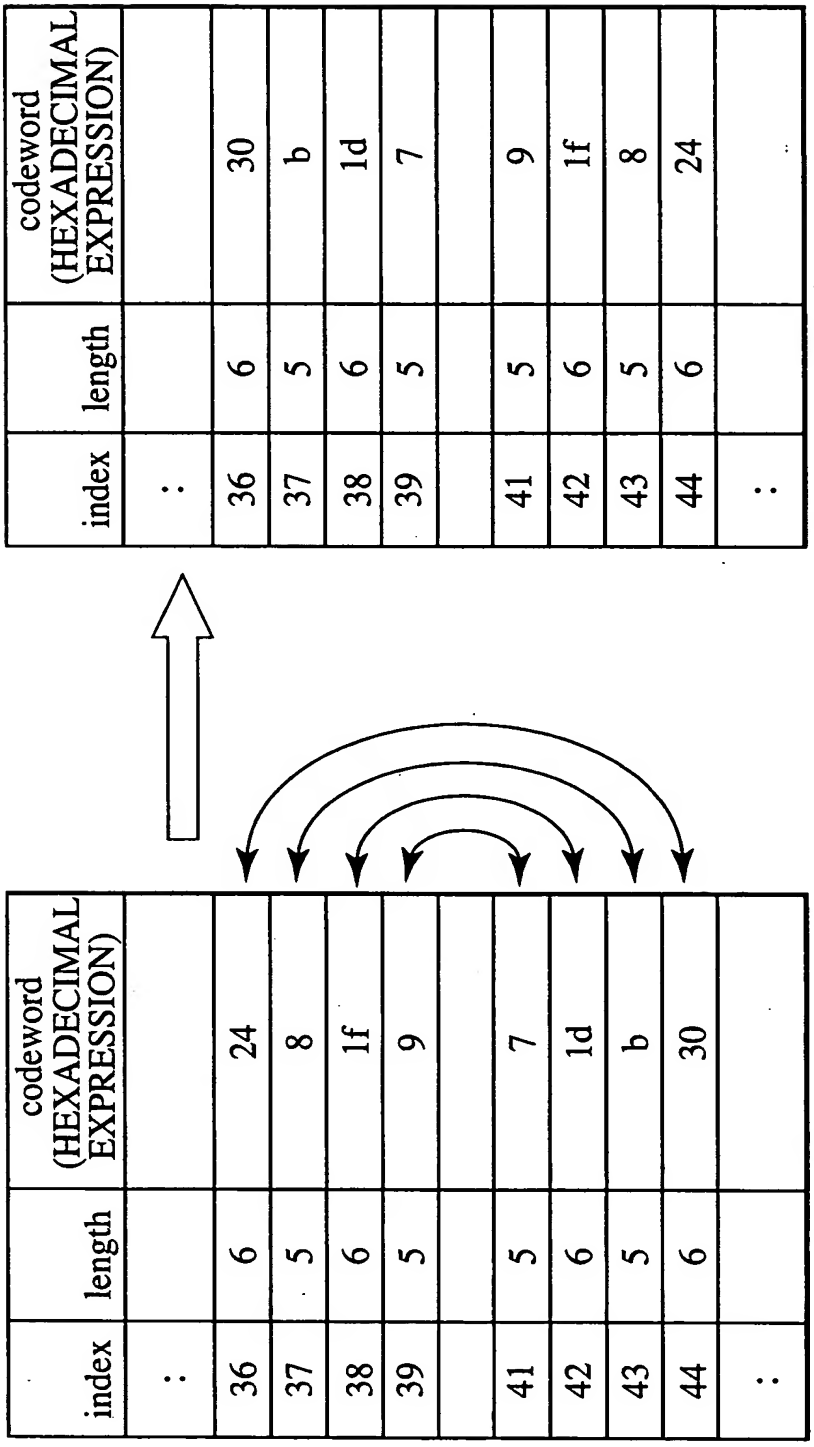
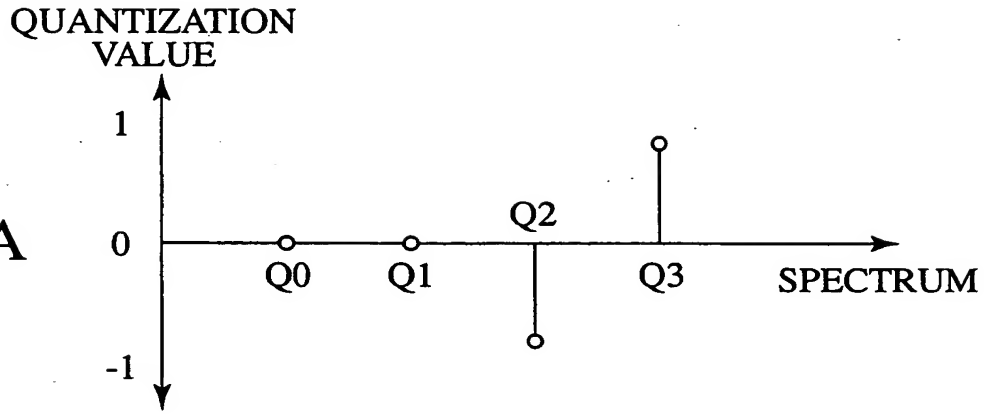

index
length
codeword (HEXADECIMAL EXPRESSION)
:
36 6 24
37 5 8
38 6 1f
39 5 9
41 5 7
42 6 1d
43 5 b
44 6 30
:
index
length
codeword (HEXADECIMAL EXPRESSION)
:
36 6 30
37 5 b
38 6 1d
39 5 7
41 5 9
42 6 1f
43 5 8
44 6 24
:

FIG.38A



index = 27×0+9×0+3×(-1)+1+40=38 → codeword:1d

codeword:1d → index = 42
= 27×0+9×0+3×1+(-1)+40

FIG.38B

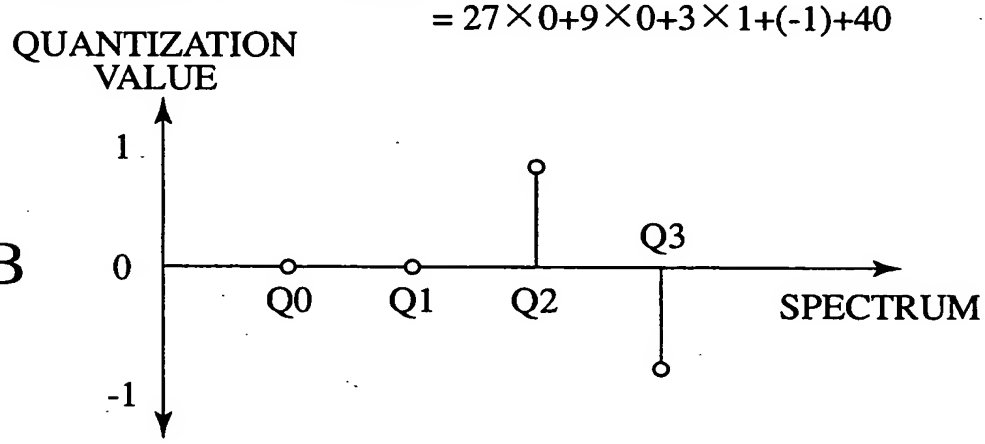


FIG.39

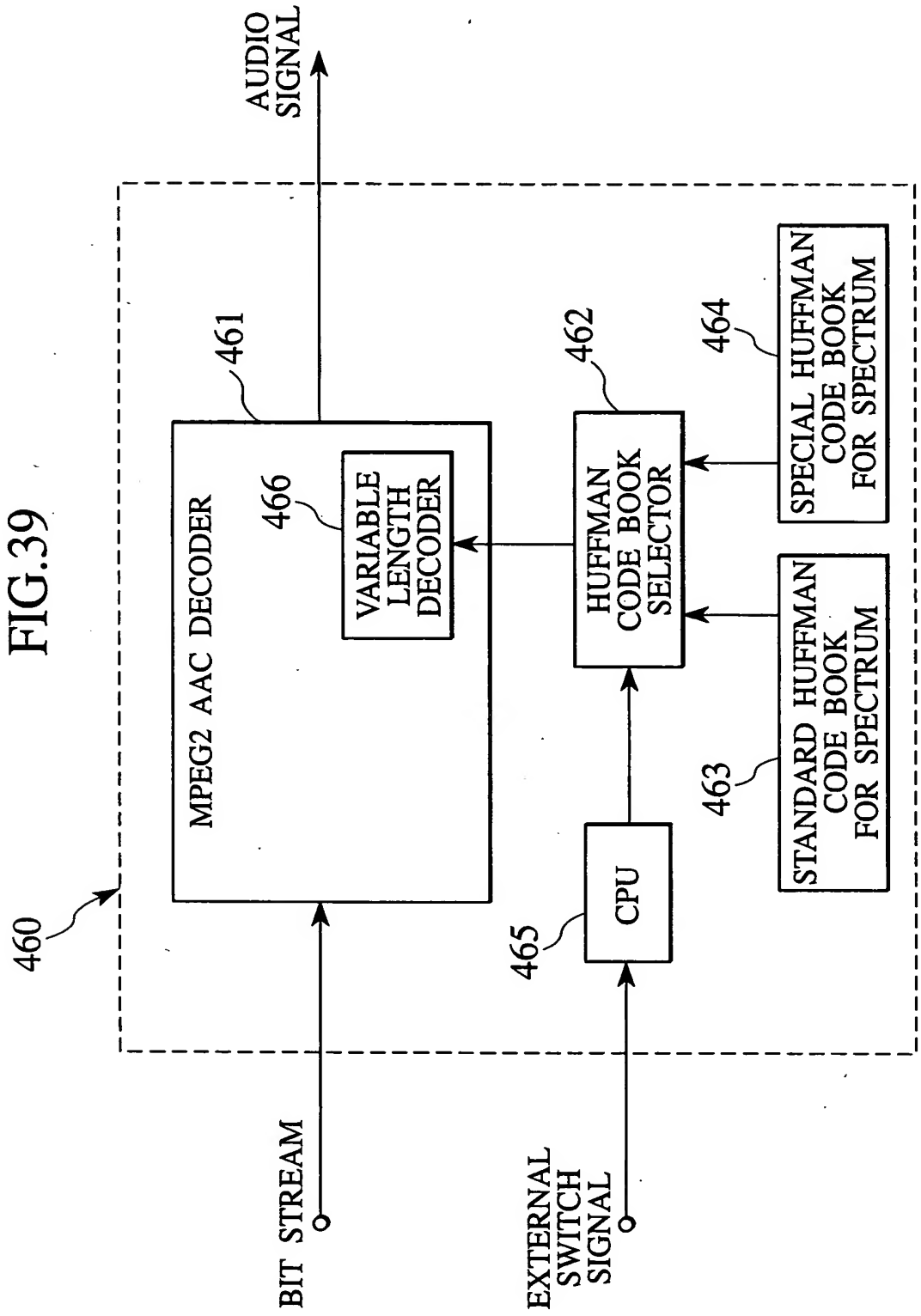

460
MPEG2 AAC DECODER
461
VARIABLE LENGTH DECODER
466
HUFFMAN CODE BOOK SELECTOR
462
SPECIAL HUFFMAN CODE BOOK FOR SPECTRUM
464
STANDARD HUFFMAN CODE BOOK FOR SPECTRUM
463
CPU
465
AUDIO SIGNAL
BIT STREAM
EXTERNAL SWITCH SIGNAL

FIG.40

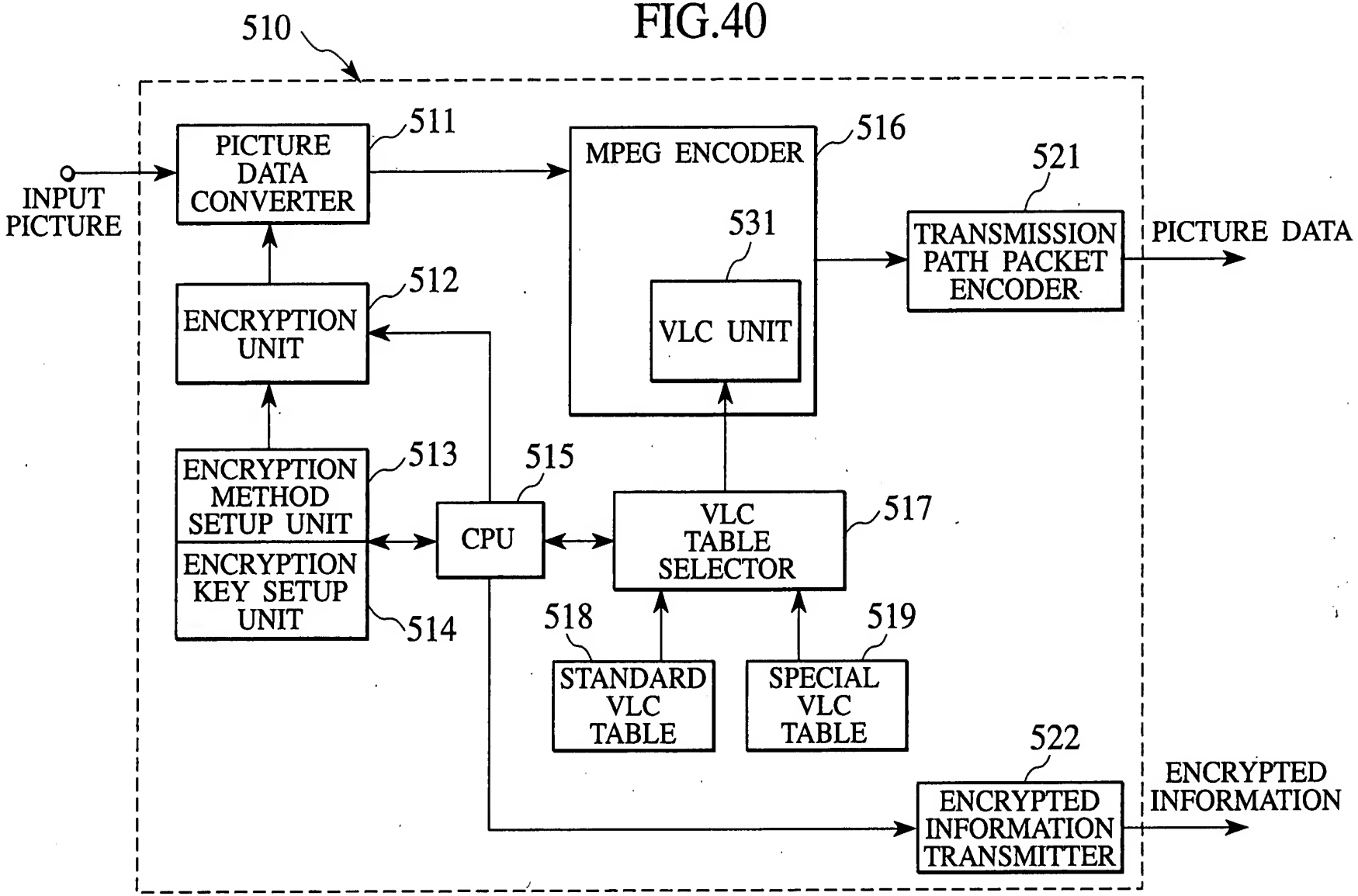

510
INPUT PICTURE
PICTURE DATA CONVERTER
511
MPEG ENCODER
516
531
VLC UNIT
521
TRANSMISSION PATH PACKET ENCODER
PICTURE DATA
ENCRYPTION UNIT
512
ENCRYPTION METHOD SETUP UNIT
513
ENCRYPTION KEY SETUP UNIT
514
CPU
515
VLC TABLE SELECTOR
517
518
STANDARD VLC TABLE
519
SPECIAL VLC TABLE
522
ENCRYPTED INFORMATION TRANSMITTER
ENCRYPTED INFORMATION

FIG.41

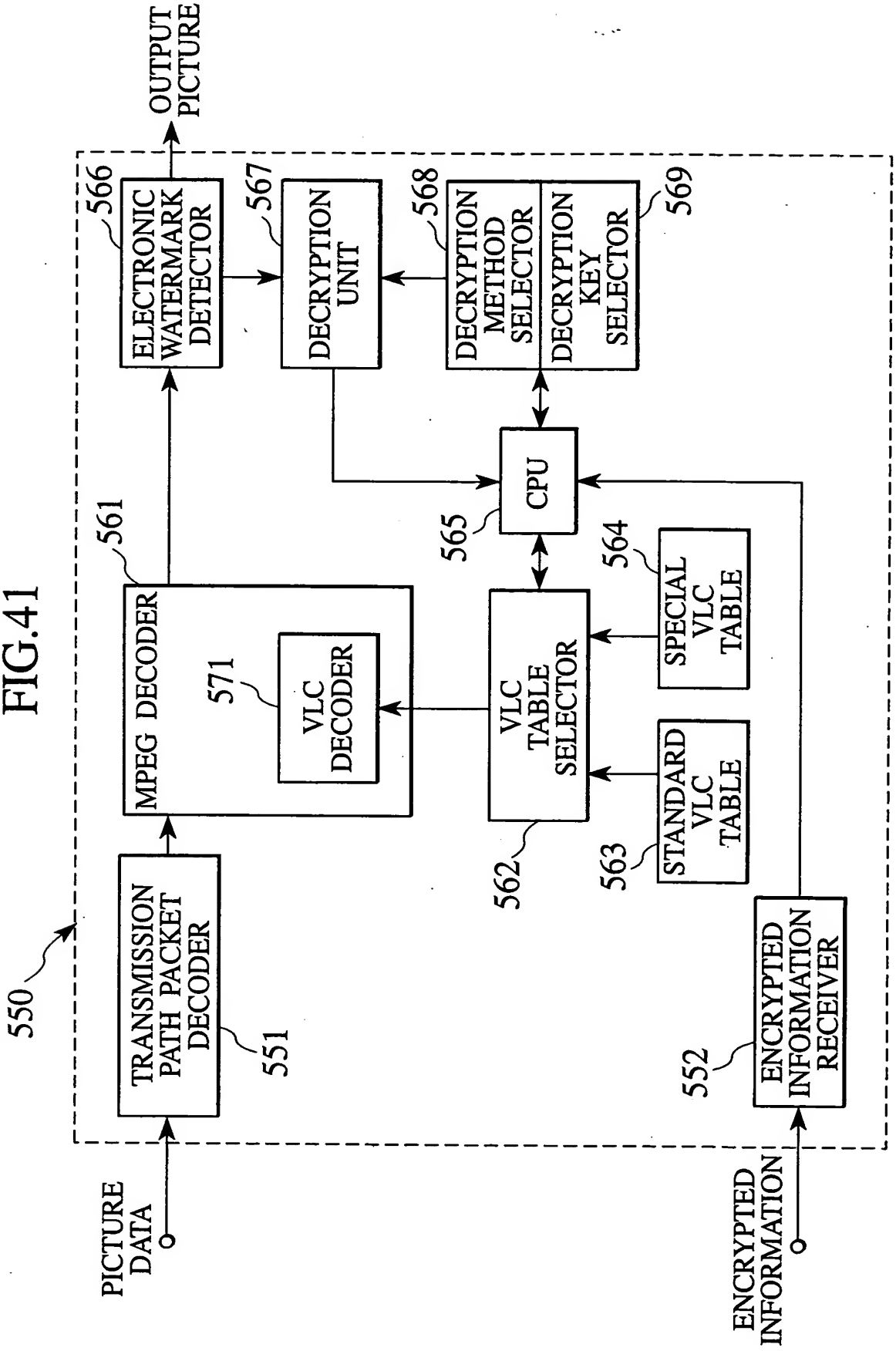

550
TRANSMISSION PATH PACKET DECODER
551
MPEG DECODER
561
571
VLC DECODER
ELECTRONIC WATERMARK DETECTOR
566
OUTPUT PICTURE
DECRYPTION UNIT
567
DECRYPTION METHOD SELECTOR
568
DECRYPTION KEY SELECTOR
569
CPU
565
VLC TABLE SELECTOR
562
SPECIAL VLC TABLE
564
STANDARD VLC TABLE
563
ENCRYPTED INFORMATION RECEIVER
552
PICTURE DATA
ENCRYPTED INFORMATION

FIG.42

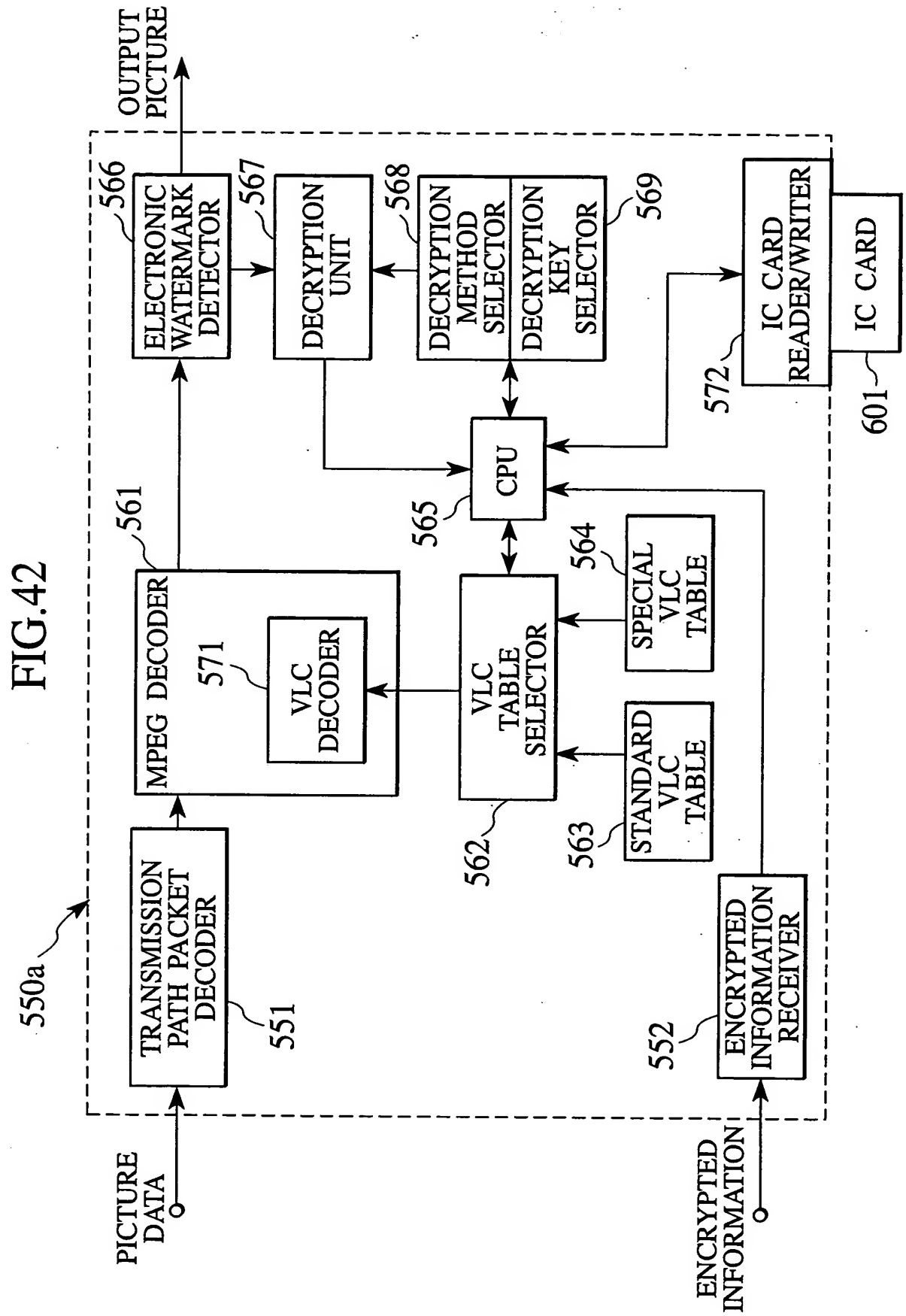

OUTPUT PICTURE
566
ELECTRONIC WATERMARK DETECTOR
567
DECRYPTION UNIT
568
DECRYPTION METHOD SELECTOR
DECRYPTION KEY SELECTOR
569
IC CARD READER/WRITER
572
IC CARD
601
CPU
565
561
MPEG DECODER
571
VLC DECODER
VLC TABLE SELECTOR
562
564
SPECIAL VLC TABLE
563
STANDARD VLC TABLE
550a
TRANSMISSION PATH PACKET DECODER
551
552
ENCRYPTED INFORMATION RECEIVER
PICTURE DATA
ENCRYPTED INFORMATION

FIG.43

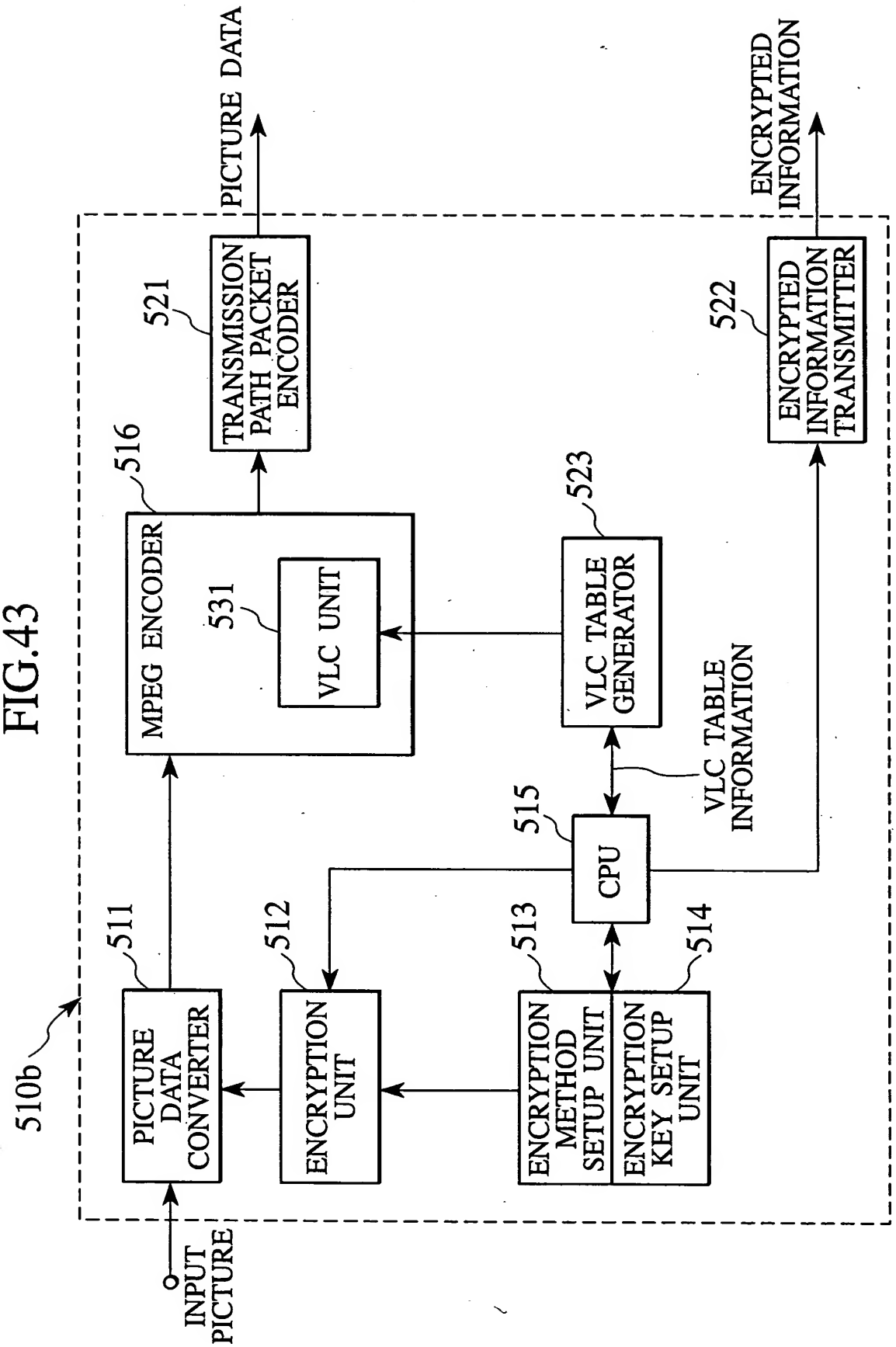

510b
INPUT PICTURE
PICTURE DATA CONVERTER
511
ENCRYPTION UNIT
512
ENCRYPTION METHOD SETUP UNIT
513
ENCRYPTION KEY SETUP UNIT
514
CPU
515
MPEG ENCODER
516
VLC UNIT
531
VLC TABLE GENERATOR
523
VLC TABLE INFORMATION
TRANSMISSION PATH PACKET ENCODER
521
PICTURE DATA
ENCRYPTED INFORMATION TRANSMITTER
522
ENCRYPTED INFORMATION

FIG.44

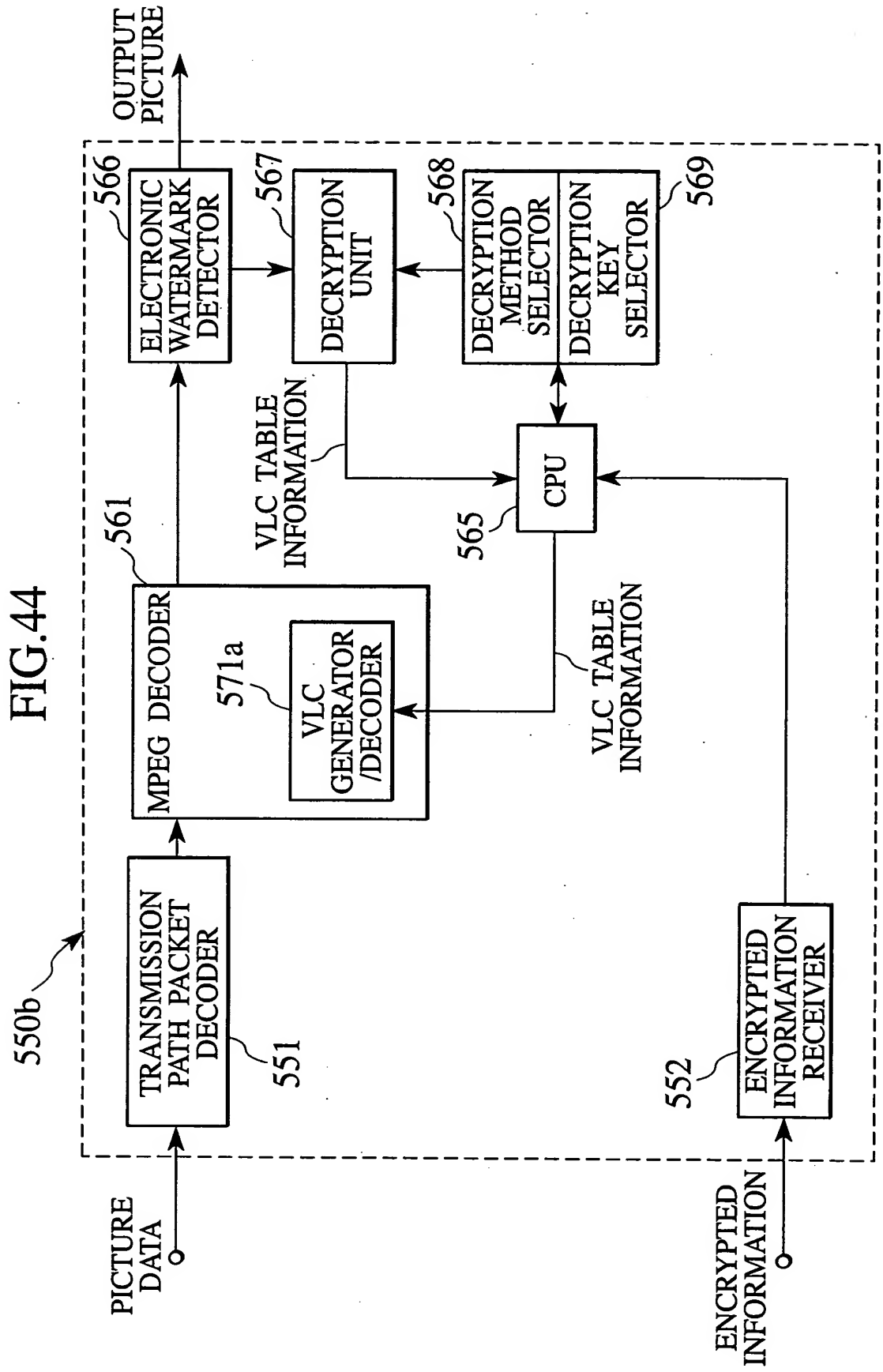

550b
PICTURE DATA
TRANSMISSION PATH PACKET DECODER
551
MPEG DECODER
561
571a
VLC GENERATOR /DECODER
VLC TABLE INFORMATION
ELECTRONIC WATERMARK DETECTOR
566
OUTPUT PICTURE
DECRYPTION UNIT
567
DECRYPTION METHOD SELECTOR
568
DECRYPTION KEY SELECTOR
569
CPU
565
VLC TABLE INFORMATION
ENCRYPTED INFORMATION RECEIVER
552
ENCRYPTED INFORMATION

FIG.45

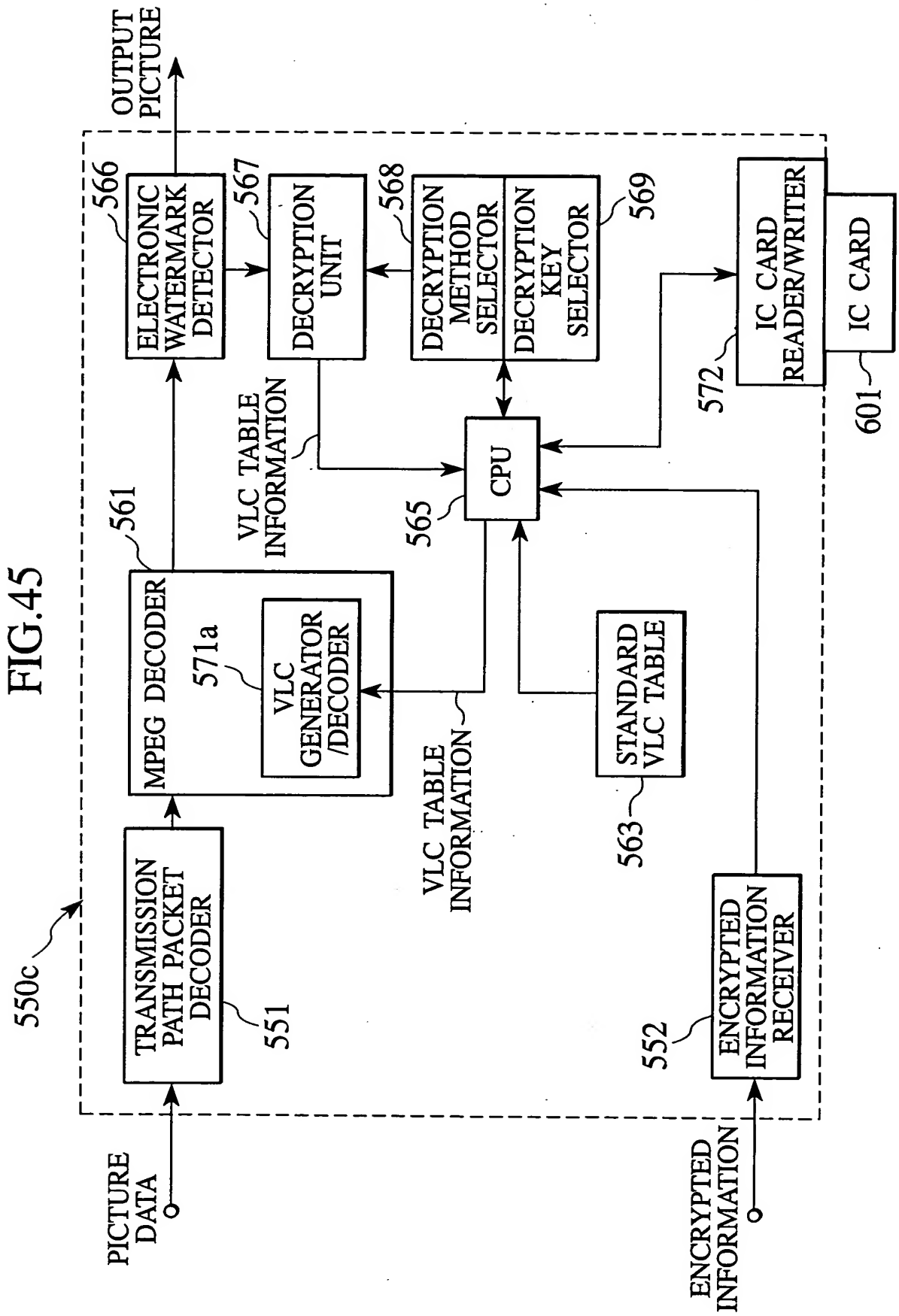

OUTPUT PICTURE
ELECTRONIC WATERMARK DETECTOR
566
DECRYPTION UNIT
567
DECRYPTION METHOD SELECTOR
568
DECRYPTION KEY SELECTOR
569
IC CARD READER/WRITER
572
IC CARD
601
VLC TABLE INFORMATION
CPU
565
561
MPEG DECODER
571a
VLC GENERATOR /DECODER
VLC TABLE INFORMATION
STANDARD VLC TABLE
563
550c
TRANSMISSION PATH PACKET DECODER
551
552
ENCRYPTED INFORMATION RECEIVER
PICTURE DATA
ENCRYPTED INFORMATION

FIG.46

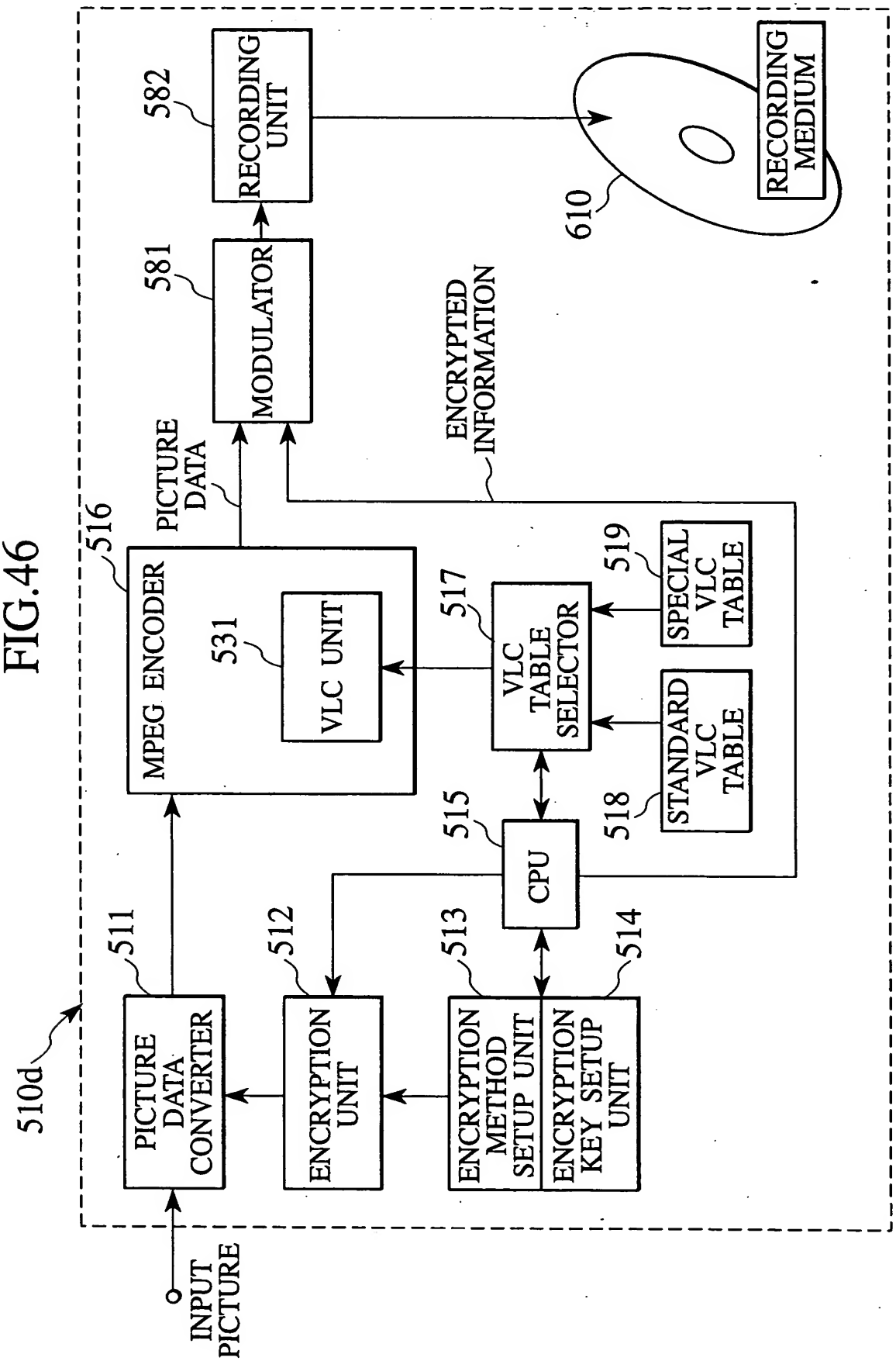

510d
INPUT PICTURE
PICTURE DATA CONVERTER
511
ENCRYPTION UNIT
512
ENCRYPTION METHOD SETUP UNIT
ENCRYPTION KEY SETUP UNIT
513
514
CPU
515
MPEG ENCODER
516
VLC UNIT
531
VLC TABLE SELECTOR
517
STANDARD VLC TABLE
518
SPECIAL VLC TABLE
519
PICTURE DATA
ENCRYPTED INFORMATION
MODULATOR
581
RECORDING UNIT
582
610
RECORDING MEDIUM

FIG.47

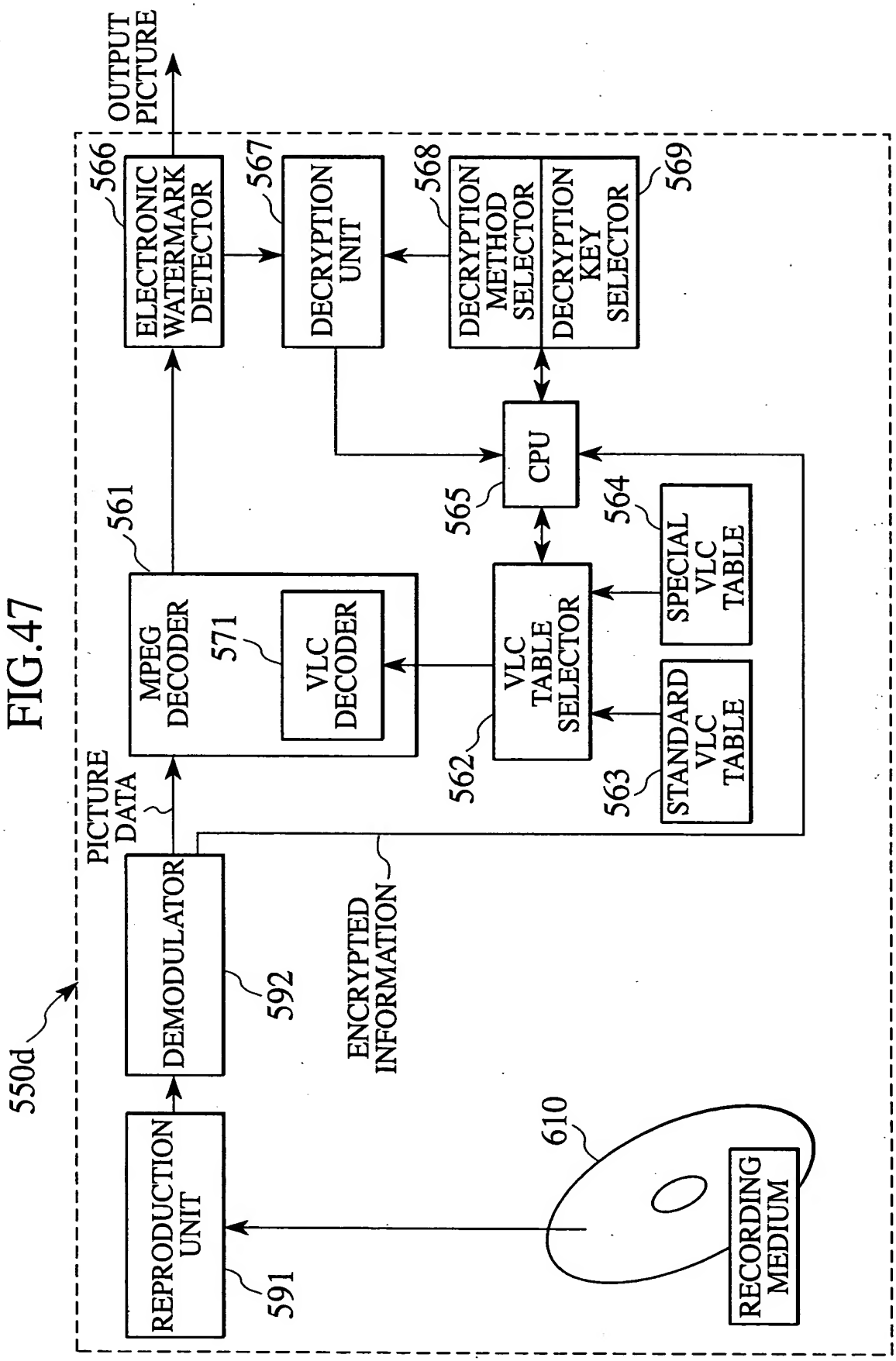

550d
REPRODUCTION UNIT
591
DEMODULATOR
592
PICTURE DATA
ENCRYPTED INFORMATION
610
RECORDING MEDIUM
MPEG DECODER
561
VLC DECODER
571
VLC TABLE SELECTOR
562
STANDARD VLC TABLE
563
SPECIAL VLC TABLE
564
CPU
565
ELECTRONIC WATERMARK DETECTOR
566
DECRYPTION UNIT
567
DECRYPTION METHOD SELECTOR
568
DECRYPTION KEY SELECTOR
569
OUTPUT PICTURE

FIG.48

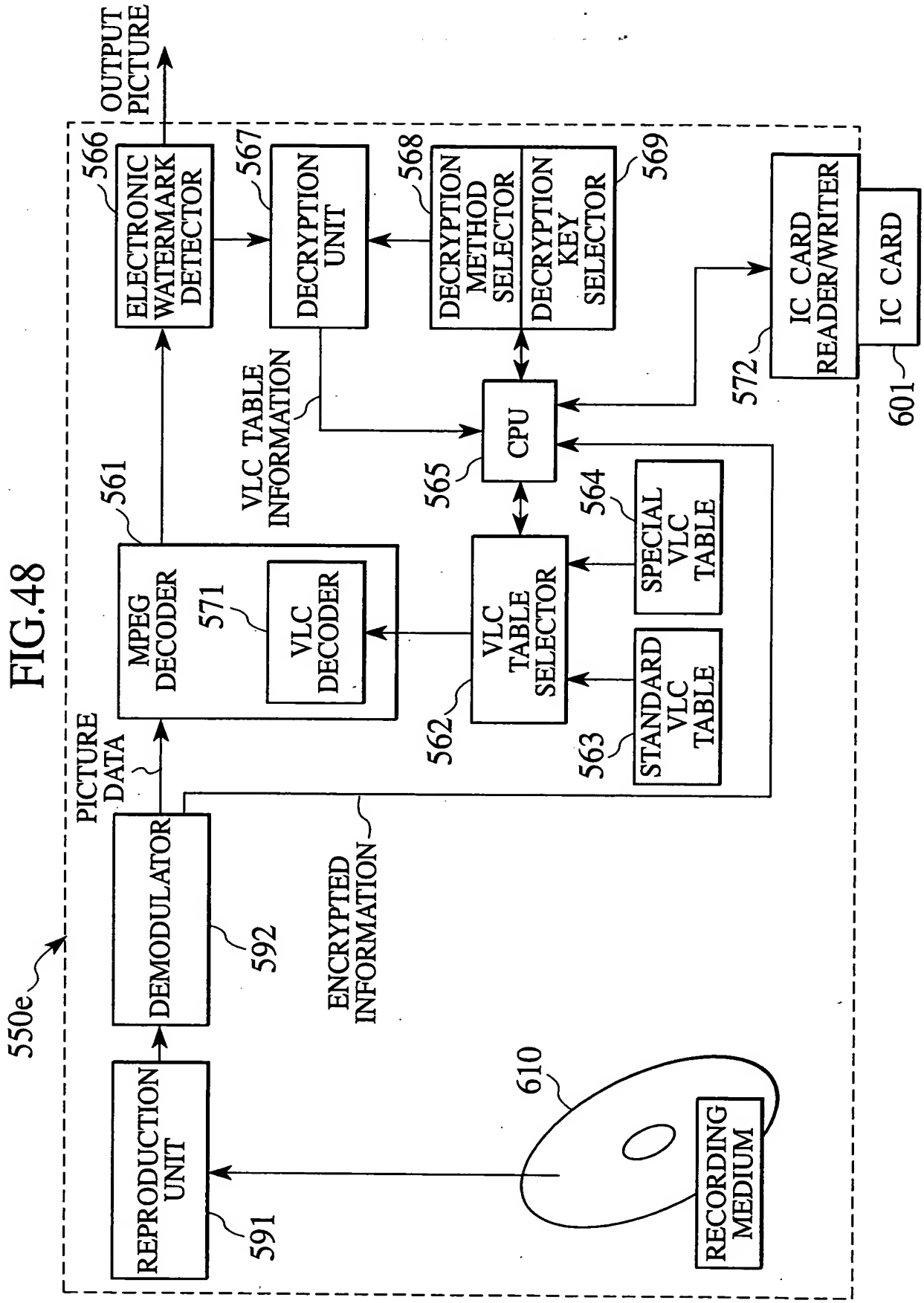

550e
REPRODUCTION UNIT
591
DEMODULATOR
592
PICTURE DATA
ENCRYPTED INFORMATION
610
RECORDING MEDIUM
MPEG DECODER
561
VLC DECODER
571
VLC TABLE SELECTOR
562
STANDARD VLC TABLE
563
SPECIAL VLC TABLE
564
CPU
565
VLC TABLE INFORMATION
ELECTRONIC WATERMARK DETECTOR
566
DECRYPTION UNIT
567
DECRYPTION METHOD SELECTOR
568
DECRYPTION KEY SELECTOR
569
OUTPUT PICTURE
IC CARD READER/WRITER
572
IC CARD
601

FIG.49

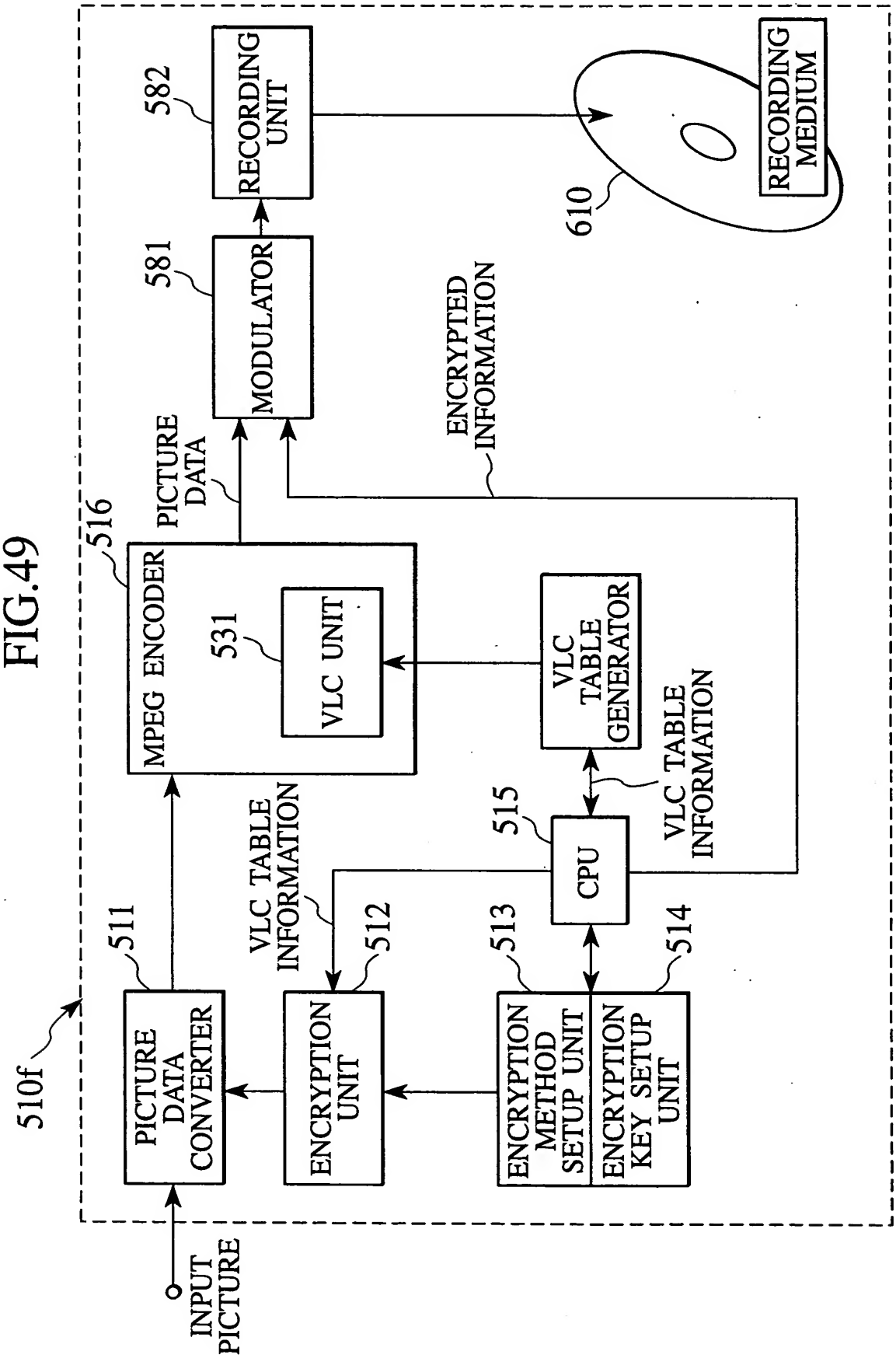

510f
INPUT PICTURE
PICTURE DATA CONVERTER
511
ENCRYPTION UNIT
512
VLC TABLE INFORMATION
ENCRYPTION METHOD SETUP UNIT
513
ENCRYPTION KEY SETUP UNIT
514
CPU
515
VLC TABLE GENERATOR
VLC TABLE INFORMATION
MPEG ENCODER
516
VLC UNIT
531
PICTURE DATA
MODULATOR
581
ENCRYPTED INFORMATION
RECORDING UNIT
582
610
RECORDING MEDIUM

FIG.50

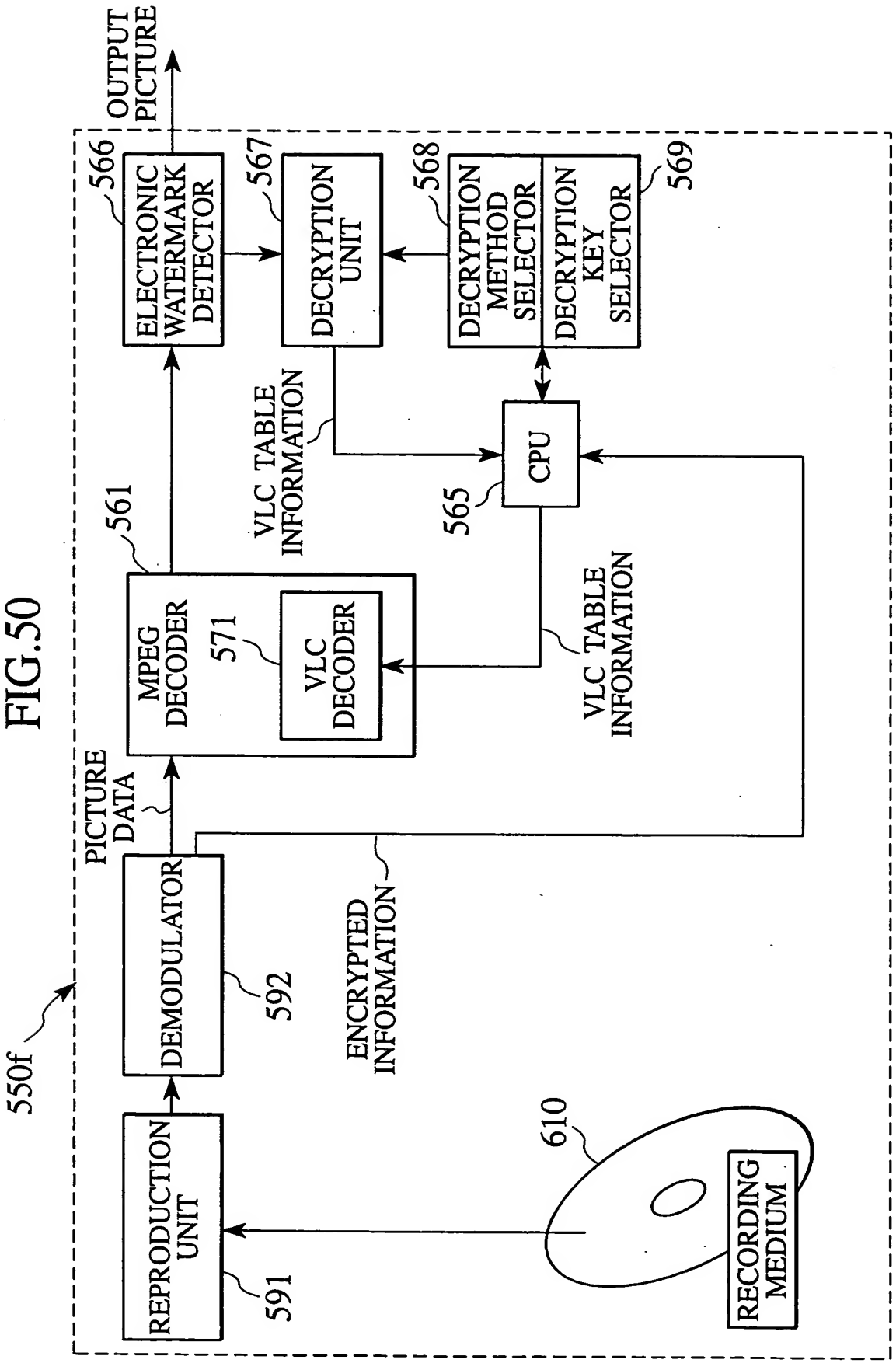
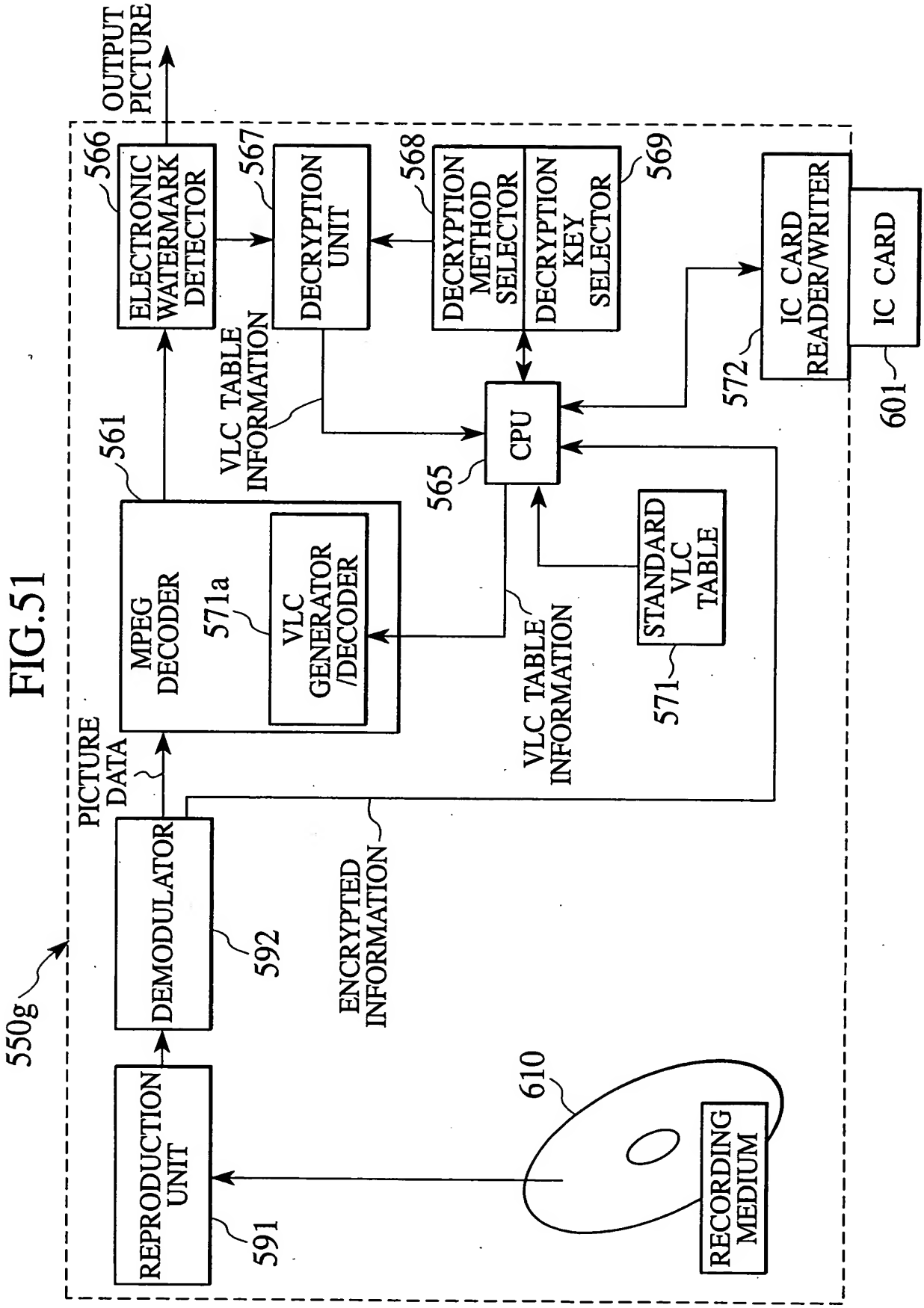

550f
REPRODUCTION UNIT
591
DEMODULATOR
592
PICTURE DATA
MPEG DECODER
561
VLC DECODER
571
ENCRYPTED INFORMATION
VLC TABLE INFORMATION
CPU
565
VLC TABLE INFORMATION
ELECTRONIC WATERMARK DETECTOR
566
DECRYPTION UNIT
567
DECRYPTION METHOD SELECTOR
568
DECRYPTION KEY SELECTOR
569
OUTPUT PICTURE
610
RECORDING MEDIUM

FIG.51


550g
REPRODUCTION UNIT
591
DEMODULATOR
592
PICTURE DATA
ENCRYPTED INFORMATION
610
RECORDING MEDIUM
MPEG DECODER
561
571a
VLC GENERATOR /DECODER
VLC TABLE INFORMATION
565
CPU
VLC TABLE INFORMATION
STANDARD VLC TABLE
571
ELECTRONIC WATERMARK DETECTOR
566
OUTPUT PICTURE
DECRYPTION UNIT
567
DECRYPTION METHOD SELECTOR
568
DECRYPTION KEY SELECTOR
569
IC CARD READER/WRITER
572
IC CARD
601